Contents

1 WS-013T00: Azure Stack HCI

- **Download Latest Student Handbook and AllFiles Content**
- **Are you a MCT?** - Have a look at our GitHub User Guide for MCTs
- **Need to manually build the lab instructions?** - Instructions are available in the MicrosoftLearning/Docker-Build repository

1.1 What are we doing?

- To support this course, we will need to make frequent updates to the course content to keep it current with the Azure services used in the course. We are publishing the lab instructions and lab files on GitHub to allow for open contributions between the course authors and MCTs to keep the content current with changes in the Azure platform.

- We hope that this brings a sense of collaboration to the labs like we've never had before - when Azure changes and you find it first during a live delivery, go ahead and make an enhancement right in the lab source. Help your fellow MCTs.

1.2 How should I use these files relative to the released MOC files?

- The instructor handbook and PowerPoints are still going to be your primary source for teaching the course content.

- These files on GitHub are designed to be used in conjunction with the student handbook, but are in GitHub as a central repository so MCTs and course authors can have a shared source for the latest lab files.

- It will be recommended that for every delivery, trainers check GitHub for any changes that may have been made to support the latest Azure services, and get the latest files for their delivery.

1.3 What about changes to the student handbook?

- We will review the student handbook on a quarterly basis and update through the normal MOC release channels as needed.

1.4 How do I contribute?

- Any MCT can submit a pull request to the code or content in the GitHub repro, Microsoft and the course author will triage and include content and lab code changes as needed.
- You can submit bugs, changes, improvement and ideas. Find a new Azure feature before we have? Submit a new demo!

1.5 Notes

1.5.1 Classroom Materials

1.6 It is strongly recommended that MCTs and Partners access these materials and in turn, provide them separately to students. Pointing students directly to GitHub to access Lab steps as part of an ongoing class will require them to access yet another UI as part of the course, contributing to a confusing experience for the student. An explanation to the student regarding why they are receiving separate Lab instructions can highlight the nature of an always-changing cloud-based interface and platform. Microsoft Learning support for accessing files on GitHub and support for navigation of the GitHub site is limited to MCTs teaching this course only.

1.7 title: Online Hosted Instructions permalink: index.html layout: home

2 Content Directory

Hyperlinks to each of the lab exercises and demos are listed below.

2.1 Labs

{% assign labs = site.pages | sort:"name" | where_exp:"page", "page.url contains '/Instructions/Labs'" %} | Module | Lab | | --- | --- | {% for activity in labs %}| {{ activity.lab.module }} | [{{ activity.lab.title }}{% if activity.lab.type %} - {{ activity.lab.type }}{% endif %}](/home/ll/Azure_clone/Azure_new/WS-013T00-Azure-Stack-HCI/{{ site.github.url }}{{ activity.url }}) | {% endfor %}

2.2 This folder is for the lab and lab answer key files.

2.3 lab: title: 'Lab: Using Windows Admin Center in hybrid scenarios' module: 'Module 2: Operating and maintaining Azure Stack HCI'

3 Lab: Using Windows Admin Center in hybrid scenarios

3.1 Scenario

Contoso, Ltd. is a medium-size financial services company with its headquarters in London, England. It currently operates almost entirely on-premises, with most of its compute environment running on the Windows Server platform, including virtualized workloads on Windows Server 2012 R2 and Microsoft Hyper-V hosts in Windows Server 2016. Its internal IT staff is well versed in Microsoft technologies, including its virtualization and software-defined datacenter offerings.

In recent months, as part of datacenter consolidation and modernization initiatives, Contoso IT migrated some of its applications to a range of Azure infrastructure as a service (IaaS) and platform as a service (PaaS) services. However, several highly regulated workloads must remain in the on-premises datacenters.

Two of these workloads present a challenge due to their performance and resiliency requirements. The first workload is a group of heavily utilized Microsoft SQL Server instances hosting transactional databases for Contoso's loan origination department. The second workload is an isolated Virtual Desktop Infrastructure (VDI) farm for users in Contoso's securities research department, which is supposed to replace an aging Windows Server 2012 R2-based Remote Desktop Services (RDS) deployment.

Contoso's Chief Information Officer (CIO) realizes that implementing these workloads will require additional investment in hardware. Before making the investment, she wants to verify that the extra expense will help the IT organization deliver a modern technological solution and accelerate the datacenter consolidation initiative.

She also wants to make sure that it helps ensure a consistent management approach that leverages existing IT skills and, if possible, integrates with some of the cloud services that Contoso is already benefiting from, such as Azure Monitor. It's also critical that the new solution provide multiple levels of high availability and resiliency, thereby protecting them from localized failures and facilitating disaster recovery to another on-premises location.

IT management has started its search for solutions that would satisfy these requirements. As lead system engineer, they have asked you to assist with searching and implementing a proof-of-concept environment that would help identify the most viable candidate.

To address the requirements for deployments of highly regulated workloads, you'll provision the core compute and networking components of the lab environment and then test integration of hyperconverged infrastructure with Azure services, including Azure Monitor and Azure Automation. You'll also test Cluster-Aware updating.

3.2 Objectives

After completing this lab, you'll be able to:

- Provision the lab environment by using PowerShell.
- Integrate hyperconverged infrastructure with Azure services.
- Review Azure integration functionality.
- Manage updates to hyperconverged infrastructure.
- Deprovision the lab environment.

3.3 Estimated time: 180 minutes

3.4 Lab setup

To connect to the lab virtual machine (VM), follow the steps the lab hosting provider provides you.

3.5 Exercise 1: Provisioning the lab environment by using PowerShell

3.5.1 Scenario

To evaluate integration of hyperconverged infrastructure with Azure, you must first provision the core compute and networking components of the lab environment.

The main tasks for this exercise are as follows:

1. Prepare the lab artifacts.
2. Deploy the lab infrastructure.

3.5.2 Task 1: Prepare the lab artifacts

1. From the lab VM, start Windows PowerShell ISE as Administrator.

2. In the Administrator: Windows PowerShell ISE window, from the console pane, run the following to remove the **Zone.Identifier** alternate data stream, which has a value of **3** indicating that it was downloaded from the Internet:

```
Get-ChildItem -Path F:\WSLab-master\ -File -Recurse | Unblock-File
```

3.5.3 Task 2: Deploy the lab infrastructure

1. On the lab VM, in the console pane of the Administrator: Windows PowerShell ISE window, set the current directory to **F:\WSLab-master\Scripts**.

2. Rename **F:\WSLab-master\Scripts\LabConfig.ps1** to **LabConfig.m2l0.ps1**.

3. Rename **F:\WSLab-master\Scripts\Scenario.ps1** to **Scenario.m2l0.ps1**.

4. Copy the **Scenario.ps1** and **Labconfig.ps1** files from **F:\WSLab-master\Scenarios\S2D and Cloud Services Onboarding** to **F:\WSLab-master\Scripts**.

5. Open the **F:\WSLab-master\Scripts\LabConfig.ps1** file, in the first line, replace **Prefix = 'WSLab-'** with **Prefix = 'WSLabOnboard-'**, and save the change.

6. From the PowerShell ISE window, run the **F:\WSLab-master\Scripts\3_Deploy.ps1** script to provision VMs for the lab environment.

> **Note**: For the Telemetry Level prompt, select the default setting of **None**. The script should complete in about seven minutes. For the prompt to start the VMs, select All to Start the VMs. When prompted with **Press enter to continue**, select **Enter**.

3.6 Exercise 2: Integrating hyperconverged infrastructure with Azure services

3.6.1 Scenario

Now that you have the core components of the lab environment provisioned, it is time to implement hyperconverged infrastructure and integrate it with Azure services.

The main tasks for this exercise are as follows:

1. Prepare the lab infrastructure VMs for integration with Azure services.
2. Provision a Storage Spaces Direct cluster within the lab environment.
3. Configure Cloud Witness quorum for the Storage Spaces Direct cluster.
4. Enable Storage Spaces Direct on the cluster.
5. Provision Azure Log Analytics workspace and Azure Log Analytics gateway.
6. Configure Azure Log Analytics workspace.
7. Integrate hyperconverged infrastructure with Azure Automation.
8. Integrate Storage Spaces Direct cluster nodes with Azure Monitor.

3.6.2 Task 1: Prepare the lab infrastructure VMs for integration with Azure services

1. On the lab VM, use PowerShell to start all the lab infrastructure VMs.

2. From the Hyper-V Manager console, connect to the **WSLabOnboard-DC** VM, and then sign in by using **CORP\LabAdmin** as the username and **LS1setup!** as the password.

3. Create a new folder **C:\Library** on the **WSLabOnboard-DC** VM.

4. Copy **F:\WSLab-master\Scripts\Scenario.ps1** from the lab VM to the folder **C:\Library** on the **WSLabOnboard-DC** VM.

5. Within the console session to the **WSLabOnboard-DC** VM, open the **C:\Library\Scenario.ps1** file in **Windows PowerShell ISE**, and then run the first part of the script marked as **#region Prereqs**.

   > **Note**: This part of the script installs prerequisites that allow subsequent parts of the script to run, including Remote Server Administration Tools and Azure PowerShell modules.

   > **Note**: Wait for the script to complete. Ignore any errors regarding **Login-AZaccount**.

6. Within the console session to the **WSLabOnboard-DC** VM, from the **Windows PowerShell ISE** window, run the second part of the script **C:\Library\Scenario.ps1** marked as **#region Install Windows Admin Center in a GW mode**.

   > **Note**: This part of the script installs Windows Admin Center in the gateway mode on the **WACGW** VM.

   > **Note**: Ignore error messages regarding aborted I/O operation.

7. Install the Microsoft Edge based on Chromium browser.

3.6.3 Task 2: Provision a Storage Spaces Direct cluster within the lab environment

1. Switch to the lab VM and from the PowerShell ISE window, shut down the VMs that will serve as nodes of the Storage Spaces Direct cluster in this lab (**WSLabOnboard-S2D1**, **WSLabOnboard-S2D2**, **WSLabOnboard-S2D3**, and **WSLabOnboard-S2D4**).

2. On the lab VM, from the PowerShell ISE window, enable nested virtualization for the VMs that will serve as nodes of the Storage Spaces Direct cluster in this lab.

3. On the lab VM, from the PowerShell ISE window, configure static memory for the VMs that will serve as nodes of the Storage Spaces Direct cluster in this lab, leaving **ProcessorCount** at **2** and setting **MemoryStartupBytes** to **4GB**.

4. On the lab VM, from the PowerShell ISE window, start the VMs that will serve as nodes of the Storage Spaces Direct cluster in this lab.

5. Switch to the console session to the **WSLabOnboard-DC** VM. From the PowerShell ISE window, install Windows Server 2019 roles and features **Hyper-V**, **Failover-Clustering**, **Data-Center-Bridging**, **RSAT-Clustering-PowerShell**, **Hyper-V-PowerShell**, and **FS-FileServer** as necessary to provision Storage Spaces Direct cluster on the four VMs in the lab environment (**S2D1**, **S2D2**, **S2D3**, and **S2D4**).

6. Within the console session to the **WSLabOnboard-DC** VM, from the PowerShell ISE window, restart **S2D1**, **S2D2**, **S2D3**, and **S2D4** VMs.

    > **Note**: Verify that the operating system in all VMs is running before you proceed to the next step.

7. Within the console session to the **WSLabOnboard-DC** VM, from the PowerShell ISE window, run the following script to configure storage to prepare for provisioning of a Storage Spaces Direct cluster on the four VMs in the lab environment (**S2D1**, **S2D2**, **S2D3**, and **S2D4**):

```
Invoke-Command ($servers) {
  Update-StorageProviderCache
  Get-StoragePool | ? IsPrimordial -eq $false | Set-StoragePool -IsReadOnly:$false -ErrorAction Si
  Get-StoragePool | ? IsPrimordial -eq $false | Get-VirtualDisk | Remove-VirtualDisk -Confirm:$fal
  Get-StoragePool | ? IsPrimordial -eq $false | Remove-StoragePool -Confirm:$false -ErrorAction Si
  Get-PhysicalDisk | Reset-PhysicalDisk -ErrorAction SilentlyContinue
  Get-Disk | ? Number -ne $null | ? IsBoot -ne $true | ? IsSystem -ne $true | ? PartitionStyle -ne
     $_ | Set-Disk -isoffline:$false
     $_ | Set-Disk -isreadonly:$false
     $_ | Clear-Disk -RemoveData -RemoveOEM -Confirm:$false
     $_ | Set-Disk -isreadonly:$true
     $_ | Set-Disk -isoffline:$true
 }
 Get-Disk | Where Number -Ne $Null | Where IsBoot -Ne $True | Where IsSystem -Ne $True | Where Par
} | Sort -Property PsComputerName, Count
```

8. Within the console session to the **WSLabOnboard-DC VM**, from the PowerShell ISE window, run cluster validation tests for **Storage Spaces Direct**, **Inventory**, **Network**, and **System Configuration** for the four VMs in the lab environment (**S2D1**, **S2D2**, **S2D3**, and **S2D4**).

    > **Note**: In order to run Test-Cluster from the **WSLabOnboard-DC** VM, you will need to install the Failover Clustering feature and restart the **WSLabOnboard-DC** VM. Ignore cluster validation errors. That's expected.

9. Within the console session to the **WSLabOnboard-DC** VM, from the PowerShell ISE window, create a new cluster named **S2DCL1** consisting of the four VMs in the lab environment (**S2D1**, **S2D2**, **S2D3**, and **S2D4**).

    > **Note**: Wait for the cluster to be provisioned.

3.6.4 Task 3: Configure Cloud Witness quorum for the Storage Spaces Direct cluster

1. Within the console session to the **WSLabOnboard-DC** VM, start the Microsoft Edge based on Chromium browser and sign in to the Azure portal using a user account with the Owner or Contributor role in the Azure subscription you will be using in this lab.

2. In the Azure portal, create a storage account with the following settings (leave others with their default values):

    Table 1: Storage account settings

Setting	Value
Subscription	the name of the Azure subscription you are using in this lab
Resource group	WS013-02-RG
Storage account name	any globally unique name between 3 and 24 in length consisting of letters and digits
Location	the name of an Azure region in proximity to the location of the lab environment
Performance	Standard
Account kind	Storage (general purpose v1)
Replication	Locally redundant storage (LRS)

> **Note**: Wait for the Storage account to be created. This should take about two minutes.

3. Copy the storage account name and its primary access key into Notepad.
4. Within the console session to the **WSLabOnboard-DC** VM, start another browser instance, navigate to the Windows Admin Center installation on the **WACGW** VM, and sign in by using **CORP\LabAdmin** as the user name and **LS1setup!** as the password.

   > **Note**: Select **Continue** if you receive an error that the connection is not secure.

5. From the **Windows Admin Center** interface, connect to the `S2DCL1.corp.contoso.com` cluster, and then authenticate by using the **CORP\LabAdmin** credentials.
6. Add **Cloud witness** quorum to the `S2DCL1.corp.contoso.com` cluster.
7. When you receive a prompt, enable CredSSP.

3.6.5 Task 4: Enable Storage Spaces Direct on the cluster

1. Within the console session to the **WSLabOnboard-DC** VM, from the PowerShell ISE window, enable Storage Spaces Direct on the newly created cluster.

   > **Note**: Disregard any error messages regarding **No disks found to be used for cache**.

2. Review the **Storage Spaces and Pools** settings of the `S2DCL1.corp.contoso.com` cluster from the **Windows Admin Center** interface.

   > **Note**: You might need to refresh the browser page to connect to the cluster.

3.6.6 Task 5: Provision Azure Log Analytics workspace and Azure Log Analytics gateway

1. Within the console session to the **WSLabOnboard-DC** VM, switch to the browser window displaying the Azure portal, start a PowerShell session in **Cloud Shell**, and use it to register the **Microsoft.Insights** and **Microsoft.AlertsManagement** resource providers.
2. Use Cloud Shell to verify that the registration was successful.

   > **Note**: Wait until the registration completes.

3. Within the console session to the **WSLabOnboard-DC** VM, in the **Windows PowerShell ISE** window, in the **C:\Library\Scenario.ps1** file in the script pane, in line **77**, replace **OutpuMode** with **OutputMode**.

   > **Note**: Run **Install-Module AZ** on the **WSLabOnboard-DC** VM prior to performing the next step.

4. Within the console session to the **WSLabOnboard-DC** VM, from the **Windows PowerShell ISE** window, run the fourth part of the script marked **#region Connect to Azure and create Log Analytics workspace if needed**.

   > **Note**: This part of the script creates the Log Analytics workspace.

5. Follow prompts to authenticate to an Azure subscription.

   > **Note**: If the user account is associated with multiple Azure subscriptions, the script will automatically display a grid with the list of your subscriptions. Select the one you want to use in this lab and then select **OK**.

   > **Note**: If you have existing Azure Log Analytics workspaces in the Azure subscription that you select, the script will automatically display a grid with the list of available Log Analytics workspaces in the Azure subscription you selected. Select **Cancel**, and the script will automatically provision one with a name that consists of the **WSLabWinAnalytics** prefix followed by the Azure subscription ID.

6. When you receive a prompt to select the Azure regions where the Log Analytics workspace will reside, select **eastus**.

   > **Note**: Make sure to select **eastus** as the target Azure region. The Azure Log Analytics location and the corresponding Azure Automation account locations must follow mappings documented in Supported regions for linked Log Analytics workspace.

7. Within the console session to the **WSLabOnboard-DC** VM, from the **Windows PowerShell ISE** window, run the fifth part of the script marked as **#region setup Log Analytics Gateway**.

   Note: This part of the script installs Log Analytics Gateway.

   Note: Disregard the warning about breaking changes to the cmdlet **Get-AzOperationalInsightsWorkspaceSh**

3.6.7 Task 6: Configure Azure Log Analytics workspace

1. Within the console session to the **WSLabOnboard-DC** VM, switch back to the browser window displaying the Azure portal.
2. In the Azure portal, navigate to the blade displaying the newly created Log Analytics workspace.

   Note: The workspace name has the **WSLabWorkspace** prefix.

3. From the Log Analytics workspace blade, enable collecting data from the **System** and **Application** Windows event logs as well as the **Processor(*)\% Processor Time** Windows performance counters.

3.6.8 Task 7: Integrate hyperconverged infrastructure with Azure Automation

1. Within the console session to the **WSLabOnboard-DC** VM, in the PowerShell ISE window, in the **C:\Library\Scenario.ps1** file in the script pane, replace line **163** with the following code:

   ```
   $location = 'eastus2'
   ```

   Note: This ensures that the location of the Azure Automation account maps to the location of the Azure Log Analytics workspace, as documented in Supported regions for linked Log Analytics workspace.

2. Within the console session to the **WSLabOnboard-DC** VM, from the **Windows PowerShell ISE** window, run the sixth part of the script marked as **#region deploy a Windows Hybrid Runbook Worker**.

   Note: This part of the script creates an Azure Automation Account and configures Hybrid Runbook Worker on the **HRWorker01** VM.

   Note: Disregard the warning about breaking changes to the cmdlet **Get-AzOperationalInsightsWorkspaceSh**

   Note: Disregard error messages during registration of the Hybrid Runbook Worker. You can verify that the registration was successful by switching to the browser displaying the Azure portal interface, navigating to the **WSLabAutomationAccount** Azure Automation account you created in this task, selecting **Hybrid worker groups**, and finally selecting the **System hybrid worker groups**. There you will find the `HRWorker01.Corp.contoso.com` entry, representing the newly registered Hybrid Runbook worker.

3. Within the console session to the **WSLabOnboard-DC** VM, in the PowerShell ISE window, in the **C:\Library\Scenario.ps1** file, replace line **286** with the following code:

   ```
   $location = 'eastus2'
   ```

4. Within the console session to the **WSLabOnboard-DC** VM, from the **Windows PowerShell ISE** window, run the seventh part of the script marked as **#region configure Hybrid Runbook Worker Addresses and Azure Automation Agent Service URL on Log Analytics Gateway**.

   Note: This part configures the Log Analytics Gateway to connect to the Azure Automation endpoints.

3.6.9 Task 8: Integrate Storage Spaces Direct cluster nodes with with Azure Monitor

1. Within the console session to the **WSLabOnboard-DC** VM, from the **Windows PowerShell ISE** window, run the eighth part of the script marked as **#region download and deploy MMA Agent to S2D cluster nodes**.

   Note: This part installs the Log Analytics agent to Storage Spaces Direct cluster nodes.

   Note: Disregard the warning about breaking changes to the cmdlet **Get-AzOperationalInsightsWorkspaceSh**

2. Within the console session to the **WSLabOnboard-DC** VM, from the **Windows PowerShell ISE** window, run the ninth part of the script marked as **#region download and install dependency agent (for service map solution)**.

Note: This part installs the Dependency agent which provides the Service Map functionality.

3.7 Exercise 3: Reviewing Azure integration functionality

3.7.1 Scenario

With the integration in place, you now must validate its capabilities by reviewing references to hyperconverged infrastructure in Azure Monitor, Log Analytics, and Azure Automation.

The main tasks for this exercise are as follows:

1. Review Log Analytics functionality.
2. Review Azure Automation functionality.
3. Review Service Map functionality.

3.7.2 Task 1: Review Log Analytics functionality

1. Within the console session to the **WSLabOnboard-DC** VM, switch back to the browser window displaying the Azure portal.

2. In the Azure portal, navigate to the blade displaying the newly created Log Analytics workspace.

   Note: The workspace name has the **WSLabWorkspace** prefix.

3. From the Log Analytics workspace blade, run the **Top 10 Virtual Machines by CPU utilization** sample query.

   Note: Review the query and the results.

   Note: The query might result in the syntax error message if the data has not been collected yet. If so, wait for a few minutes and try again or return to this task once you complete the rest of the lab.

3.7.3 Task 2: Review Azure Automation functionality

1. Within the console session to the **WSLabOnboard-DC** VM, in the browser window displaying the Azure portal, navigate back to the blade displaying the Log Analytics workspace you were reviewing in the previous task.

2. On the Log Analytics workspace blade, in the **Related Resources** section, select **Automation Account**.

3. Note the information regarding the linked Automation account, and then select **Go to account**.

4. On the **WSLabAutomationAccount** blade, in the **Configuration Management** section, select **Inventory** and note that you have the option to **Enable** the Inventory solution.

5. Without making any changes, on the **WSLabAutomationAccount** blade, in the **Configuration Management** section, select **Change tracking**.

6. On the **Change tracking** blade, note that you have the option to **Enable** the Change tracking solution.

7. Without making any changes, on the **WSLabAutomationAccount** blade, in the **Process automation** section, select **Hybrid worker groups**.

8. On the **Hybrid worker groups** blade, select the **System hybrid worker groups** tab and note that it contains a separate group for each server that was registered with Azure Automation, with a single worker per group.

   Note: Verify that last seen time for each worker group is within one hour of the current time.

3.7.4 Task 3: Review Service Map functionality

1. Within the console session to the **WSLabOnboard-DC** VM, in the browser window displaying the Azure portal, navigate back to the blade displaying the Log Analytics workspace you were reviewing in the first task of this exercise.

2. On the Log Analytics workspace blade, in the **General** section, select **Workspace summary**.

3. On the **Overview** blade, review the list of solutions that you implemented in the previous exercise and navigate to the **Service Map** blade.

> **Note**: It may take several minutes for the **Service Map** blade to appear.

4. On the **Service Map** blade, on the **Machines** tab, in the list of monitored servers, select **S2D1** (one of the nodes of the Storage Spaces Direct cluster), zoom into the diagram in the center of the blade, and then review the **Summary** pane on the right-hand side of the blade.

5. With the **S2D1** server selected, display each of the sections on the right-hand side of the pane, including **Summary**, **Properties**, **Alerts**, **Log Events**, **Performance**, **Security**, and **Updates**.

   > **Note**: In the **Security** section, if you find the **Logons with a clear text password** entry, select it. You will be automatically redirected to the Log Analytics workspace blade displaying the corresponding Kusto Query Language (KQL) query.

6. With the **S2D1** server selected, zoom in further on the diagram, and then expand the rectangle representing the **S2D2** cluster node.

   > **Note**: Review the list of connections, and verify that they involve multiple processes (such as **clussvc** and **System**).

7. Review the diagram and note that it includes connections to `DC.corp.contoso.com` over ports 53 (dns), 67 (bootps), 88, 123, 135, 389, and 445 (there might be others).

   > **Note**: These connections are listed even though the **DC** server does not have the Log Analytics and Dependency agents installed.

3.8 Exercise 4: Managing updates to hyperconverged infrastructure

3.8.1 Scenario

One of your objectives is to determine the most efficient approach to deploying updates to servers that are part of your hyperconverged environment. You decided to evaluate the use of Cluster Aware Updating and Azure Automation Update Management.

The main tasks for this exercise are as follows:

1. Implement Cluster Aware Updating by using Windows Admin Center.
2. Use Azure Automation update management.

3.8.2 Task 1: Implement Cluster Aware Updating by using Windows Admin Center

1. Within the console session to the **WSLabOnboard-DC** VM, switch back to the **Windows Admin Center** interface, and then in the list of **Tools** of the `S2DCL1.corp.contoso.com` page, select **Updates**.

2. From the **Windows Admin Center** interface, enable **Cluster Aware Updating**, and then check for available updates.

3. Review the list of available updates without making any changes.

   > **Note**: You have the option to **Apply All Updates**. Do not select it.

   > **Note**: You can monitor the status of applying the updates directly from the **Cluster Aware Updating** panel.

3.8.3 Task 2: Use Azure Automation update management

1. Within the console session to the **WSLabOnboard-DC** VM, switch back to the browser window displaying the **WSLabAutomationAccount** blade in the Azure portal.

2. From the **WSLabAutomationAccount** blade, navigate to the **Update Management** blade.

3. From the **Update Management** blade, review the list of machines and identify noncompliant ones.

   > **Note**: To schedule an update deployment, you must first create a computer group.

4. Within the console session to the **WSLabOnboard-DC** VM, in the browser displaying the Azure portal, navigate back to the blade displaying the Log Analytics workspace you were reviewing in the previous exercise.

5. On the Log Analytics workspace blade, navigate to the list of example queries.

6. From the list of example queries, load the **Missing security or critical updates** from the **Virtual Machine** section into the editor window.
7. In the editor window, remove the line '| summarize count() by Classification', select **Run** and review results of the query.

   Note: the query lists all of missing security or critical updates.

8. In the editor window, replace the query with the following one:

```
Update
| where UpdateState == 'Needed' and Optional == false and Classification == 'Security Updates' and
| distinct Computer
```

   Note: Computer group queries must use the `distinct Computer` clause.

   Note: The query excludes servers which are members of the Storage Spaces Direct cluster since these are updated by using Cluster Aware Updating.

9. Run the query to verify that the query returns the list of noncompliant servers in the `Corp.contoso.com` domain that are not part of the Storage Spaces Direct cluster.
10. Save the query with the following settings:

Table 2: Group query settings

Setting	Value
Name	`corp.contoso.com non-compliant non S2D servers`
Save as	Function
Function Alias	`corp_non_s2d_non_compliant`
Save this query as a computer group	Enabled
Category	Updates

11. In the Azure portal, navigate back to the **Update Management** blade of the **WSLabAutomationAccount** Automation Account.
12. From the **WSLabAutomationAccount** Automation Account blade, schedule an update deployment with the following settings (leave others with their default values):

Table 3: Update deployment settings

Setting	Value
Name	ws01302 update deployment
Operating system	Windows
Groups to update	`corp.contoso.com non-compliant non S2D servers`
Machines to update	`corp.contoso.com non-compliant non S2D servers`
Schedule settings	date and time at least 5 minutes ahead of the current date and time

   Note: You can use the **Include/exclude updates** setting to include or exclude individual updates.

   Note: You can use the **Pre-scripts + Post-scripts** setting to specify scripts to run before and after patch deployment.

13. Back on the **Update management** blade, navigate to the **Deployment schedules** tab, and ensure that the deployment has been successfully scheduled.

3.9 Exercise 5: Deprovisioning the lab environment

3.9.1 Scenario

To minimize Azure-related charges, you'll deprovision the Azure resources provisioned throughout this lab. You'll also revert the state of the lab environment to its original state in preparation for further testing.

The main tasks for this exercise are as follows:

1. Deprovision the Azure resources.
2. Deprovision the lab resources.

3.9.2 Task 1: Deprovision the Azure resources

1. Switch to the lab VM.
2. Start a browser, navigate to the Azure portal, and then sign in with the Owner or Contributor role in the Azure subscription you will be using in this lab.
3. Within the Azure portal, start a PowerShell session in Cloud Shell.
4. From the Cloud Shell pane, run the following to remove all Azure resources you provisioned in this lab:

```
Get-AzResourceGroup -Name 'WS013-02-RG' | Remove-AzResourceGroup -Force -AsJob
Get-AzResourceGroup -Name 'WSLabWinAnalytics' | Remove-AzResourceGroup -Force -AsJob
```

3.9.3 Task 2: Deprovision the lab resources

1. On the lab VM, start Windows PowerShell ISE as Administrator.
2. From the PowerShell ISE window, run the **F:\WSLab-master\Scripts\Cleanup.ps1** script to remove all VMs provisioned in this lab.

3.9.4 Results

3.10 After completing this lab, you will have provisioned the lab environment by using PowerShell, integrated hyperconverged infrastructure with Azure services, reviewed Azure integration functionality, managed updates to hyperconverged infrastructure, and deprovisioned the lab environment.

3.11 lab: title: 'Lab A: Implementing a Storage Spaces Direct cluster by using Windows PowerShell' module: 'Module 3: Planning for and implementing Azure Stack HCI Storage'

4 Lab A: Implementing a Storage Spaces Direct cluster by using Windows PowerShell

4.1 Scenario

One of your objectives is to minimize the effort associated with deployment and management of on-premises resources. As part of this effort, you want to test the process of implementing a Storage Spaces Direct cluster in an automated manner by using Windows PowerShell.

4.2 Objectives

After completing this lab, you'll be able to implement a Storage Spaces Direct cluster by using Windows PowerShell.

4.3 Estimated time: 25 minutes

4.4 Lab setup

To connect to the virtual machine (VM) for the lab, follow the steps provided to you by the lab hosting provider.

4.5 Exercise 1: Implementing a Storage Spaces Direct cluster by using Windows PowerShell

4.5.1 Scenario

To minimize the effort associated with deployment and management of on-premises resources, you will test the process of implementing a Storage Spaces Direct cluster in an automated manner by using Windows PowerShell.

The main tasks for this exercise are as follows:

1. Provision the lab environment VMs.

2. Configure the management server.
3. Deploy a Storage Spaces Direct cluster on the lab VMs by using PowerShell.

4.5.2 Task 1: Provision the lab environment VMs

1. On the lab VM, in the **Administrator: Windows PowerShell ISE** window, from the **script** pane, run the following command to rename **LabConfig.ps1** and **Scenario.ps1**:

```
Set-Location -Path 'F:\WSLab-master\Scripts'
Move-Item -Path '.\LabConfig.ps1' -Destination '.\LabConfig.m312.ps1' -Force
Move-Item -Path '.\Scenario.ps1' -Destination '.\Scenario.m312.ps1' -Force
```

2. On the lab VM, in the **Administrator: Windows PowerShell ISE** window, from the **script** pane, save the following command as **F:\WSLab-master\Scripts\LabConfig.ps1**:

```
$LabConfig=@{ DomainAdminName = 'LabAdmin'; AdminPassword = 'LS1setup!'; Prefix = 'WSLab-'; Secure
1..4 | % {
     $VMNames = "S2D";
     $LABConfig.VMs += @{
     VMName = "$VMNames$_" ;
     Configuration = 'S2D' ;
     ParentVHD = 'Win2019Core_G2.vhdx';
     HDDNumber = 12;
     HDDSize = 4TB ;
     MemoryStartupBytes = 4GB;
     NestedVirt = $True
    }
}
$LabConfig.VMs += @{
     VMName = 'Management' ;
     Configuration = 'Simple';
     ParentVHD = 'Win2019_G2.vhdx';
     StaticMemory = $true;
     MemoryStartupBytes = 8GB;
     AddToolsVHD = $True;
     DisableWCF = $True;
     VMProcessorCount = 4
}
```

3. On the lab VM, in the **Administrator: Windows PowerShell ISE** window, from the **script** pane, run the **F:\WSLab-master\Scripts\3_Deploy.ps1** script to provision VMs for the Storage Spaces Direct environment.

    Note: For the Telemetry prompt select **None**. The script should complete in about 10 minutes.

4. After the script completes, in the **Administrator: Windows PowerShell ISE** window, from the **console** pane, run the following command to start the newly provisioned VMs that will host the Storage Spaces Direct environment:

```
Get-VM -Name 'WSLab-Management' | Start-VM
Start-Sleep 150
Get-VM | Where-Object Name -like 'WSLab-S2D*' | Start-VM -AsJob
```

5. On the **console** pane of the **Administrator: Windows PowerShell ISE** window, run the following command to copy the **Scenario.ps1** file from **F:\WSLab-master\Scenarios\S2D Hyperconverged** to the current directory:

```
Copy-Item -Path 'F:\WSLab-master\Scenarios\S2D Hyperconverged\Scenario.ps1' -Destination '.\'
```

6. On the lab VM, start **Hyper-V Manager** and connect via a console session to **WSLab-DC**. When prompted to sign in, provide the username **CORP\LabAdmin** and the password **LS1setup!**.

7. In the **WSLab-DC** VM console session, start **Windows PowerShell ISE** as an administrator.

8. From the **Administrator: Windows PowerShell ISE** window, run `slmgr -rearm` and then select **OK**.

9. From the **Administrator: Windows PowerShell ISE** window, run `Restart-Computer -Force`.

   Note: Make sure that the **WSLab-DC VM** is running before you proceed to the next task.

4.5.3 Task 2: Configure the management server

1. On the lab VM, from **Hyper-V Manager** and connect with a console session to **WSLab-Management**. When prompted to sign in, provide the username **CORP\LabAdmin** and password **LS1setup!**.
2. In the **WSLab-Management** VM console session, start Windows PowerShell ISE as Administrator.
3. In the **WSLab-Management** VM console session, from the **Administrator: Windows PowerShell ISE** window, run the following command to install Remote Server Administration Tools:

   ```
   Install-WindowsFeature -Name RSAT-Clustering,RSAT-Clustering-Mgmt,RSAT-Clustering-PowerShell,RSAT-
   ```

   Note: Proceed to the next step without waiting for the installation to complete.
4. In the **WSLab-Management** VM console session, start another instance of Windows PowerShell ISE as Administrator.
5. In the **WSLab-Management** VM console session, from the newly started **Administrator: Windows PowerShell ISE** window, run the following command to download and install the Microsoft Edge (Chromium) browser:

   ```
   $progressPreference='SilentlyContinue'
   Invoke-WebRequest -Uri "https://go.microsoft.com/fwlink/?linkid=2069324&language=en-us&Consent=1"
   Start-Process -FilePath "$env:USERPROFILE\Downloads\MicrosoftEdgeSetup.exe" -Wait
   ```

   Note: Proceed to the next step without waiting for the installation to complete.
6. In the **WSLab-Management** VM console session, start another instance of Windows PowerShell ISE as Administrator.
7. In the **WSLab-Management** VM console session, from the newly started **Administrator: Windows PowerShell ISE** window, run the following command to download and install Windows Admin Center:

   ```
   Invoke-WebRequest -UseBasicParsing -Uri https://aka.ms/WACDownload -OutFile "$env:USERPROFILE\Down
   Start-Process msiexec.exe -Wait -ArgumentList "/i $env:USERPROFILE\Downloads\WindowsAdminCenter.ms
   ```

   Note: Proceed to the next step without waiting for the installation to complete.
8. Switch to the first **Administrator: Windows PowerShell ISE** window where you initiated the installation of the Remote Server Administration Tools, wait for the installation to complete, and then from the **script** pane, run the following command to configure Kerberos constrained delegation to minimize prompts for credentials when using Windows Admin Center:

   ```
   $gateway = "Management"
   $nodes = Get-ADComputer -Filter * -SearchBase "ou=workshop,DC=corp,dc=contoso,DC=com"
   $gatewayObject = Get-ADComputer -Identity $gateway
   foreach ($node in $nodes){
    Set-ADComputer -Identity $node -PrincipalsAllowedToDelegateToAccount $gatewayObject
   }
   ```

   Note: Before you proceed to the next step, verify that the installation of Microsoft Edge and Windows Admin Center completed.
9. Close the other two instances of the **Administrator: Windows PowerShell ISE** window you opened earlier in this task.
10. Switch to the Microsoft Edge browser window, navigate to `https://management.corp.contoso.com`, and when prompted to authenticate, sign in as **CORP\LabAdmin** with **LS1setup!** as the password.

4.5.4 Task 3: Deploy a Storage Spaces Direct cluster on the lab VMs by using PowerShell

1. In the **WSLab-Management** VM console session, start File Explorer and create a **C:\Library** folder.
2. Copy **F:\WSLab-master\Scripts\Scenario.ps1** from the lab VM to the **C:\Library** folder in the **WSLab-Management** VM.

3. In the **WSLab-Management** VM console session, from the **Administrator: Windows PowerShell ISE** window, run the **C:\Library\Scenario.ps1** script.

   > **Note:** Wait for the script to complete before you proceed to the next lab. The script should complete in about 35 minutes.

4.6 Results

4.7 After completing this lab, you will have initiated PowerShell-based deployment of a Storage Spaces Direct cluster.

4.8 lab: title: 'Lab B: Managing storage of a Storage Spaces Direct cluster by using Windows Admin Center and Windows PowerShell' module: 'Module 3: Planning for and implementing Azure Stack HCI Storage'

5 Lab B: Managing storage of a Storage Spaces Direct cluster by using Windows Admin Center and Windows PowerShell

5.1 Scenario

Now that you have provisioned a Storage Spaces Direct cluster in an automated manner by using Windows PowerShell, you want to determine whether you can minimize administrative effort associated with remediating disk failures in a Storage Spaces Direct cluster by leveraging its resiliency and self-healing capabilities.

5.2 Objectives

After completing this lab, you'll be able to manage storage of a Storage Spaces Direct cluster by using Windows Admin Center and Windows PowerShell.

5.3 Estimated time: 50 minutes

5.4 Lab setup

To connect to the virtual machine (VM) for the lab, follow the steps provided to you by the lab hosting provider.

5.5 Exercise 1: Managing storage of a Storage Spaces Direct cluster by using Windows Admin Center and Windows PowerShell

You want to determine whether you can minimize the administrative effort associated with remediating disk failures in a Storage Spaces Direct cluster. You will do this by examining its resiliency and self-healing capabilities using Windows Admin Center and Windows PowerShell.

The main tasks for this exercise are as follows:

1. Review the installation of the Storage Spaces Direct cluster on the lab VMs.
2. Create and manage volumes by using Windows Admin Center.
3. Review the health status of the Storage Spaces Direct cluster.
4. Simulate removing a disk from the Storage Spaces Direct cluster.
5. Simulate returning a disk to the Storage Spaces Direct cluster.
6. Simulate removing a disk and replacing it with a different one.
7. Deprovision the lab resources.

> **Note:** Ensure that the **Scenario.ps1** script you started in the previous lab has completed successfully before you start this exercise.

5.5.1 Task 1: Review the installation of the Storage Spaces Direct cluster on the lab VMs

1. In the **WSLab-Management** VM console session, in the browser window displaying Windows Admin Center, connect to the `s2d-cluster.corp.contoso.com` cluster, and authenticate as **CORP\LabAdmin** with the password **LS1setup!**.

2. In the browser window displaying Windows Admin Center, on the `s2d-cluster.corp.contoso.com` page, navigate to the inventory of the Storage Spaces Direct cluster volumes.

3. Review the list of volumes and verify that each of them is listed with the **OK** status.

4. In the browser window displaying Windows Admin Center, on the `s2d-cluster.corp.contoso.com` page, navigate to the inventory of the Storage Spaces Direct cluster drives.

5. Review the list of drives and verify that each of them is listed with the **OK** status.

6. In the browser window displaying the Windows Admin Center interface, on the `s2d-cluster.corp.contoso.com` page, review the virtual switch configuration. Note that each node is connected to an external switch named **SETSwitch**. Review the list of network adapters attached to the switch and verify that the load balancing algorithm is set to **Hyper-V port**.

7. In the browser window displaying Windows Admin Center, from the `s2d-cluster.corp.contoso.com` page, display the **Settings** panel, and verify that the cluster contains a single storage pool.

   > **Note:** You have the option of assigning an arbitrary name to the storage pool.

8. On the **Settings** panel, select **Storage Spaces Direct** and review cache settings.

   > **Note:** The **Cache mode for HDD** is set by default to **Read/Write** and **Cache mode for SSD** is set to **Write only**. You have the option of modifying these settings.

9. In the browser window displaying the Windows Admin Center interface, on the `s2d-cluster.corp.contoso.com` page, on the **Settings** panel, in the **Cluster** section, review the following entries:

   - Access point. Verify that **Cluster name** is set to **S2D-Cluster**.
   - Node shutdown behavior. Verify that the setting **Move virtual machines on node shutdown** is enabled.
   - Cluster traffic encryption. Verify that **Core traffic** is set to **Sign** and **Storage traffic** to **Clear text**.
   - Virtual machine load balancing. Verify that **Balance virtual machines** is set to **Always** with **Aggressiveness** set to **Low**.
   - Witness. Verify that **Witness type** is set to **File share witness** with **File share path** set to **\\DC\S2D-ClusterWitness**.

10. In the **WSLab-Management** VM console session, start **Failover Cluster Manager**.

11. In **Failover Cluster Manager**, connect to the `s2d-cluster.corp.contoso.com` cluster, review the list of disks, verify that the cluster contains a single pool named **Cluster Pool 1**, and examine the storage pool properties, including virtual and physical disks.

12. In **Failover Cluster Manager**, select the **Networks** node and note that it contains separate entries for **Management** and **SMB** networks, with two network adapters on each cluster node.

5.5.2 Task 2: Create and manage volumes by using Windows Admin Center

1. In the browser window displaying Windows Admin Center, navigate to the panel listing inventory of volumes on the **s2d-cluster** Storage Spaces Direct cluster.

   > **Note:** The inventory at this point includes only pre-created **ClusterPerformanceHistory** volume.

2. From the panel listing inventory of volumes on the **s2d-cluster** Storage Spaces Direct cluster, create a new volume with the settings listed in the following table:

   Table 1: Three-way volume settings

Setting	Value
Name	Volume01-3wm
Resiliency	Three-way mirror
Size on HDD	100
Size units	GB

   > **Note:** Review the resulting estimated footprint and the total available storage space.

3. From the panel listing inventory of volumes on the **s2d-cluster** Storage Spaces Direct cluster, repeat the same sequence of steps to configure the settings of a new volume as listed in the following table:

   Table 2: Mirror-accelerated parity volume settings

Setting	Value
Name	Volume02-map70
Resiliency	Mirror-accelerated parity
Parity percentage	70% parity, 30% mirror
Size on HDD	100
Size units	GB

4. Review the resulting estimated footprint and the total available storage space and then select **More options**.

5. In the **More options** section, note the message indicating that to use deduplication and compression, it's necessary to install the **Data Deduplication** role on every server.

6. Create the volume without enabling deduplication and compression.

7. In the **WSLab-Management** VM console session, from the **script** pane of the **Administrator: Windows PowerShell ISE** window, run the following command to install the **Data Deduplication** Windows Server role service on each cluster node:

   ```
   $servers = @('S2D1','S2D2','S2D3','S2D4')
   Invoke-Command -ComputerName $servers -ScriptBlock {Install-WindowsFeature -Name FS-Data-Deduplica
   ```

   > **Note:** Wait for the installation to complete before you proceed to the next step. This should take about three minutes.

8. Switch to the browser window displaying the Windows Admin Center interface and enable deduplication on the **Volume02-map70** volume with the **Hyper-V Deduplication mode**.

   > **Note:** You might need to close and re-open the browser page displaying the Windows Admin Center interface to account for the installation of the **Data Deduplication** role service.

9. From the panel displaying configuration of the **Volume02-map70** volume, expand its size to **200 GB**.

10. Review the settings of the volume **Volume02-map70**, including **Optional features**, and verify that it contains **Dual parity** and **Three-way mirror** under **Storage Tiers**.

    > **Note:** You have the option of enabling or disabling encryption and compression, but it's not possible to modify integrity checksum or resiliency settings after the volume is created.

11. From the **Volume02-map70** pane in Windows Admin Center, navigate to the content of **C: > ClusterStorage > Volume02-map70**.

    > **Note:** The Windows Admin Center automatically displays the page with the content of **C: > ClusterStorage > Volume02-map70** on the Hyper-V cluster node, which serves as the owner of the corresponding volume.

12. Copy the **tools.vhdx** file from the **F:\WSLab-master\Scripts\ParentDisks** folder on the lab VM to the **Downloads** folder on the **WSLab-Management** VM.

13. Switch to the console session connected to the **WSLab-Management** VM, and in the browser window displaying Windows Admin Center, create a new folder **C:\ClusterStorage\Volume02-map70\vhdFiles**, and then upload the **tools.vhdx** file into it.

5.5.3 Task 3: Review the health status of the Storage Spaces Direct cluster

1. In the **WSLab-Management** VM console session, from the **Administrator: Windows PowerShell ISE** window, run the following command to identify the health status of the Storage Spaces Direct cluster:

   ```
   $storagesubsystem = Get-StorageSubSystem -CimSession s2d-cluster -FriendlyName Cl*
   $storagesubsystem
   ```

   > **Note:** Ensure that the **HealthStatus** is listed as **Healthy** and **OperationalStatus** is listed as **OK**.

2. In the **WSLab-Management** VM console session, switch to the browser window displaying Windows Admin Center, and on the `s2d-cluster.corp.contoso.com` page, review the content of **Dashboard** and verify that it reports **Healthy** status for all cluster components.

3. In the **WSLab-Management** VM console session, switch to the **Administrator: Windows PowerShell ISE** window and then run the following command to identify any health faults of the Storage Spaces Direct cluster:

   ```
   Get-HealthFault -CimSession s2d-cluster
   ```

   Note: If there are no health faults, the cmdlet should return **WARNING: s2d-cluster: There aren't any faults right now**.

4. In the **WSLab-Management** VM console session, from the **Administrator: Windows PowerShell ISE** window, run the following command to identify the health actions of the Storage Spaces Direct cluster.

   ```
   $storagesubsystem | Get-StorageHealthAction -CimSession s2d-cluster
   ```

   Note: Verify that there are no pending health actions.

5. In the **WSLab-Management** VM console session, from the **Administrator: Windows PowerShell ISE** window, run the following command to identify the status of virtual disks of the Storage Spaces Direct cluster:

   ```
   Get-VirtualDisk -CimSession s2d-cluster | Sort-Object FriendlyName
   ```

   Note: Ensure that the **HealthStatus** is listed as **Healthy** and **OperationalStatus** as **OK**.

5.5.4 Task 4: Simulate removing a disk from the Storage Spaces Direct cluster

1. Switch to the lab VM, and from the **Administrator: Windows PowerShell ISE** window, run the following command to choose a random disk in one of the nodes of the S2D cluster:

   ```
   $diskToPull = Get-VM -Name WSLab-s2d* | Get-VMHardDiskDrive | Where-Object ControllerLocation -ge
   $diskToPull
   ```

2. On the lab VM, from the **Administrator: Windows PowerShell ISE** window, run the following command to simulate removal of the disk you identified in the previous step:

   ```
   $pulledDiskPath = $diskToPull.Path
   $diskToPull | Remove-VMHardDiskDrive
   ```

3. Switch to the console session connected to the **WSLab-Management** VM, and from the **Administrator: Windows PowerShell ISE** window, rerun the following command to identify the status of the virtual disks of the Storage Spaces Direct cluster:

   ```
   Get-VirtualDisk -CimSession s2d-cluster | Sort-Object FriendlyName
   ```

   Note: Verify that **HealthStatus** is listed as **Warning** and **OperationalStatus** as **Incomplete** for all virtual disks.

4. In the **WSLab-Management** VM console session, from the **Administrator: Windows PowerShell ISE** window, run the following command to identify the health status of the Storage Spaces Direct disks:

   ```
   Get-PhysicalDisk -CimSession s2d-cluster
   ```

   Note: Review the output of the cmdlet and note that the **Operational Status** of one of the disks is listed as **Lost Communication**.

5. In the **WSLab-Management** VM console session, from the **Administrator: Windows PowerShell ISE** window, rerun the following command to identify the health status of the Storage Spaces Direct cluster:

   ```
   Get-HealthFault -CimSession s2d-cluster
   ```

   Note: Review the faults displayed by the Health service.

6. In the **WSLab-Management** VM console session, switch to the browser window displaying Windows Admin Center, navigate to the volume summary pane of the `s2d-cluster.corp.contoso.com` Storage Spaces Direct cluster and review the listing of alerts.

7. In the **WSLab-Management** VM console session, in the browser window displaying Windows Admin Center, navigate to the volume inventory pane of the `s2d-cluster.corp.contoso.com` Storage Spaces Direct cluster.

8. Review the list of volumes and identify the ones which are listed with the **Needs repair** status.

9. In the **WSLab-Management** VM console session, in the browser window displaying Windows Admin Center, navigate to the drive summary pane of the `s2d-cluster.corp.contoso.com` Storage Spaces Direct cluster and review the listing of alerts.

10. In the **WSLab-Management** VM console session, in the browser window displaying Windows Admin Center, navigate to the drive inventory pane of the `s2d-cluster.corp.contoso.com` Storage Spaces Direct cluster.

11. In the drive inventory, identify the drive with the **Lost communication** status.

12. In the browser window displaying Windows Admin Center, on the `s2d-cluster.corp.contoso.com` page, navigate to the **Dashboard** pane and review its contents, verifying that it includes alerts indicating a drive issue.

5.5.5 Task 5: Simulate returning a disk to the Storage Spaces Direct cluster

1. Switch to the lab VM, and from the **Administrator: Windows PowerShell ISE** window, run the following command to simulate returning the disk removed in the previous task to the same node of the Storage Spaces Direct cluster:

   ```
   Add-VMHardDiskDrive -VMName $disktopull.VMName -Path $pulledDiskPath
   ```

2. Switch to the console session connected to the **WSLab-Management** VM, and from the **Administrator: Windows PowerShell ISE** window, rerun the following command to identify the status of the virtual disks of the Storage Spaces Direct cluster:

   ```
   Get-VirtualDisk -CimSession s2d-cluster | Sort-Object FriendlyName
   ```

   > **Note:** The **HealthStatus** of virtual disks should be listed again as **Healthy**, with **OperationalStatus** listed as **OK**. If you observe one of the virtual disks listed as **InService**, wait for about one minute and repeat this step.

3. In the **WSLab-Management** VM console session, from the **Administrator: Windows PowerShell ISE** window, rerun the following command to identify the health status of the Storage Spaces Direct cluster:

   ```
   Get-HealthFault -CimSession s2d-cluster
   ```

   > **Note:** The storage subsystem should return to the healthy status in about five minutes, so you might need to wait and rerun the cmdlet if you are still observing messages indicating faults.

4. In the **WSLab-Management** VM console session, switch to the browser window displaying Windows Admin Center, navigate to the **volume summary** pane of the `s2d-cluster.corp.contoso.com` Storage Spaces Direct cluster and review listing of alerts.

5. In the **WSLab-Management** VM console session, in the browser window displaying Windows Admin Center, navigate to the **volume inventory** pane of the `s2d-cluster.corp.contoso.com` Storage Spaces Direct cluster.

6. Review the list of volumes and verify that each of them is listed with the **OK** status.

7. In the **WSLab-Management** VM console session, in the browser window displaying Windows Admin Center, navigate to the **drive summary** pane of the `s2d-cluster.corp.contoso.com` Storage Spaces Direct cluster and verify that there are no alerts.

8. In the **WSLab-Management** VM console session, in the browser window displaying Windows Admin Center, navigate to the **drive inventory** pane of the `s2d-cluster.corp.contoso.com` Storage Spaces Direct cluster.

9. In the drive inventory, verify that all drives are listed with the **OK** status.

10. In the browser window displaying Windows Admin Center, on the `s2d-cluster.corp.contoso.com` page, navigate to the **Dashboard** pane and review its contents, verifying that it reports **Healthy** status for all cluster components and displays a single alert indicating the sync operation.

11. In the **WSLab-Management** VM console session, from the **script** pane of the **Administrator: Windows PowerShell ISE** window, run the following command to identify the health status of the Storage Spaces Direct cluster:

```
$storagesubsystem = Get-StorageSubSystem -CimSession s2d-cluster -FriendlyName Cl*
$storagesubsystem
```

> **Note:** Before you proceed to the next task, verify that the **HealthStatus** is listed as **Healthy** and that the **OperationalStatus** is listed as **OK**.

5.5.6 Task 6: Simulate removing a disk and replacing it with a different one

1. Switch to the lab VM, and from the **Administrator: Windows PowerShell ISE** window, run the following command to choose a random disk from one of the nodes of the Storage Spaces Direct cluster:

   ```
   $diskToPull = Get-VM -Name WSLab-s2d* | Get-VMHardDiskDrive | Where-Object ControllerLocation -ge
   $diskToPull
   ```

2. On the lab VM, from the **Administrator: Windows PowerShell ISE** window, run the following command to simulate removal of the disk you identified in the previous step:

   ```
   $pulledDiskPath = $diskToPull.Path
   $diskToPull | Remove-VMHardDiskDrive
   ```

3. On the lab VM, from the **Administrator: Windows PowerShell ISE** window, run the following command to simulate replacing the removed disk with another one:

   ```
   $newDiskPath = "$(($pulledDiskPath).Substring(0,$pulledDiskPath.Length-5))_NEW.vhdx"
   New-VHD -Path $newDiskPath -SizeBytes 4TB
   Add-VMHardDiskDrive -VMName $diskToPull.VMName -Path $newDiskPath
   ```

4. Switch to the console session connected to the **WSLab-Management** VM, and from the **Administrator: Windows PowerShell ISE** window, rerun the following command to identify the status of the virtual disks of the Storage Spaces Direct cluster:

   ```
   Get-VirtualDisk -CimSession s2d-cluster | Sort-Object FriendlyName
   ```

   > **Note:** Verify that the **OperationalStatus** for virtual disks is listed **Incomplete** and **HealthStatus** is listed as **Warning**.

5. In the **WSLab-Management** VM console session, from the **Administrator: Windows PowerShell ISE** window, rerun the following command to verify that the repair and regeneration jobs are in progress on the Storage Spaces Direct cluster:

   ```
   $storagesubsystem = Get-StorageSubSystem -CimSession s2d-cluster -FriendlyName Cl*
   $storagesubsystem | Get-StorageJob
   ```

   > **Note:** If you don't observe any jobs, wait one minute and rerun the cmdlets.

6. In the **WSLab-Management** VM console session, from the **Administrator: Windows PowerShell ISE** window, run the following command to identify the health status of the Storage Spaces Direct disks:

   ```
   Get-PhysicalDisk -CimSession s2d-cluster
   ```

   > **Note:** Review the output of the cmdlet and note that **Operational Status** of one of the disks is listed as **{Removing From Pool, Lost Communication}** and **Usage** is listed as **Retired**.

7. In the **WSLab-Management** VM console session, from the **Administrator: Windows PowerShell ISE** window, run the following command to identify the retired disk:

   ```
   Get-PhysicalDisk -CimSession s2d-cluster -Usage retired
   ```

8. In the **WSLab-Management** VM console session, from the **Administrator: Windows PowerShell ISE** window, run the following command to identify the health status of the Storage Spaces Direct cluster.

   ```
   Get-HealthFault -CimSession s2d-cluster
   ```

9. Review the output of the cmdlet and note that the drive will be automatically retired after 15 minutes of lost communication.

10. In the **WSLab-Management** VM console session, switch to the browser window displaying Windows Admin Center, navigate to the **volume summary** pane of the `s2d-cluster.corp.contoso.com` Storage Spaces Direct cluster, and review the listing of alerts.

11. In the **WSLab-Management** VM console session, in the browser window displaying Windows Admin Center, navigate to the volume inventory pane of the `s2d-cluster.corp.contoso.com` Storage Spaces Direct cluster.
12. Review the list of volumes and verify that each of them is listed with the **Needs repair** status.
13. In the **WSLab-Management** VM console session, in the browser window displaying Windows Admin Center, navigate to the **drive summary** pane of the `s2d-cluster.corp.contoso.com` Storage Spaces Direct cluster, and review the listing of alerts.
14. In the **WSLab-Management** VM console session, in the browser window displaying Windows Admin Center, navigate to the **drive inventory** pane of the `s2d-cluster.corp.contoso.com` Storage Spaces Direct cluster.
15. In the drive inventory, identify the drive with the **Retired, Removing from pool, Lost communication** status.
16. In the browser window displaying Windows Admin Center, on the `s2d-cluster.corp.contoso.com` page, navigate to the **Dashboard** pane and review its contents, verifying that it includes alerts that indicate a drive issue.
17. In the **WSLab-Management** VM console session, switch to the **Administrator: Windows PowerShell ISE** window and run the following command to verify whether the retired disk has been automatically removed from the Storage Spaces Direct cluster:

    ```
    Get-PhysicalDisk -CimSession s2d-cluster -Usage retired
    ```

    > **Note:** Verify that the command doesn't return any output. If that's not the case, wait a few minutes and rerun the command.

18. In the **WSLab-Management** VM console session, switch to the browser window displaying Windows Admin Center, navigate to the **volume summary** pane of the `s2d-cluster.corp.contoso.com` Storage Spaces Direct cluster, and review the listing of alerts.
19. In the **WSLab-Management** VM console session, in the browser window displaying Windows Admin Center, navigate to the **volume inventory** pane of the `s2d-cluster.corp.contoso.com` Storage Spaces Direct cluster.
20. Review the list of volumes and verify that each of them is listed with the **OK** status.
21. In the **WSLab-Management** VM console session, in the browser window displaying Windows Admin Center, navigate to the **drive summary** pane of the `s2d-cluster.corp.contoso.com` Storage Spaces Direct cluster, and verify that there are no alerts.
22. In the **WSLab-Management** VM console session, in the browser window displaying Windows Admin Center, navigate to the **drive inventory** pane of the `s2d-cluster.corp.contoso.com` Storage Spaces Direct cluster.
23. In the drive inventory, verify that all drives are listed with the **OK** status.
24. In the browser window displaying Windows Admin Center, on the `s2d-cluster.corp.contoso.com` page, navigate to the **Dashboard** pane and review its contents, verifying that it reports **Healthy** status for all cluster components and a single alert indicating the sync operation.

5.5.7 Task 7: Deprovision the lab resources

1. Switch to the lab VM.
2. On the lab VM, from the **Administrator: Windows PowerShell ISE** window, run the **F:\WSLab-master\Scripts\Cleanup.ps1** script to remove all VMs provisioned in this lab.
3. In the **Administrator: Windows PowerShell ISE** window, close the tab displaying the **F:\WSLab-master\Scripts\Cleanup.ps1** script.

5.6 Results

5.7 After completing this lab, you will have managed storage in a Storage Spaces Direct cluster by using Windows Admin Center and Windows PowerShell. --- lab: title: 'Lab C: Managing and monitoring resiliency of a Storage Spaces Direct cluster' module: 'Module 3: Planning for and implementing Azure Stack HCI Storage'

6 Lab C: Managing and monitoring resiliency of a Storage Spaces Direct cluster

6.1 Scenario

You want to examine resiliency in situations when there are simultaneous cluster node and drive failures. You want to understand how resiliency can protect cluster stability and integrity. To start, you will create tiered volumes and test volume, disk, and cluster resiliency.

6.2 Objectives

After completing this lab, you'll be able to manage and monitor resiliency of a Storage Spaces Direct cluster.

6.3 Estimated time: 55 minutes

6.4 Lab setup

To connect to the VM for the lab, follow the steps provided to you by the lab hosting provider.

6.5 Exercise 1: Managing and monitoring resiliency of a Storage Spaces Direct cluster

6.5.1 Scenario

You have evaluated the self-healing capabilities of Storage Spaces Direct clusters after removing individual disks. Now you want to examine resiliency in situations when there are simultaneous cluster node and drive failures.

The main tasks for this exercise are as follows:

1. Provision the lab environment VMs.
2. Configure the management server.
3. Create and configure a failover cluster.
4. Configure fault domains on the failover cluster.
5. Enable Storage Spaces Direct on the failover cluster.
6. Review fault domain configuration on the Storage Spaces Direct cluster.
7. Create tiered volumes on a Storage Spaces Direct failover cluster.
8. Test resiliency of the Storage Spaces Direct cluster.
9. Restore failed disks and nodes on the Storage Spaces Direct cluster.
10. Deprovision the lab resources.

6.5.2 Task 1: Provision the lab environment VMs

1. On the lab VM, in the **Administrator: Windows PowerShell ISE** window, run the following command to rename **LabConfig.ps1** and **Scenario.ps1**:

```
Set-Location -Path 'F:\WSLab-master\Scripts'
Move-Item -Path '.\LabConfig.ps1' -Destination '.\LabConfig.m3l3.ps1' -Force -ErrorAction Silently
Move-Item -Path '.\Scenario.ps1' -Destination '.\Scenario.m3l3.ps1' -Force -ErrorAction SilentlyCo
```

2. In the **Administrator: Windows PowerShell ISE** window, from the **script** pane, save the following content as **F:\WSLab-master\Scripts\LabConfig.ps1**:

```
$LabConfig=@{ DomainAdminName='LabAdmin'; AdminPassword='LS1setup!'; Prefix = 'WSLab-'; SwitchName
1..6 | % {
    $VMNames="S2D";
    $LABConfig.VMs += @{
    VMName = "$VMNames$_" ;
    Configuration = 'S2D' ;
    ParentVHD = 'Win2019Core_G2.vhdx';
    SSDNumber = 0;
    SSDSize=800GB ;
    HDDNumber = 12;
    HDDSize= 4TB ;
    MemoryStartupBytes= 1GB;
    NestedVirt=$false
   }
}
$LabConfig.VMs += @{
    VMName = 'Management' ;
    Configuration = 'Simple';
    ParentVHD = 'Win2019_G2.vhdx';
    StaticMemory = $true;
    MemoryStartupBytes = 8GB;
    AddToolsVHD = $True;
    DisableWCF = $True;
    VMProcessorCount = 4
}
```

3. In the **Administrator: Windows PowerShell ISE** window, open and run the **F:\WSLab-master\Scripts\3_Deploy.ps1** script to provision VMs for the Storage Spaces Direct environment.

   Note: Select **None** at the Telemetry prompt. The script should complete in about 10 minutes.

4. When the script completes, in the **Administrator: Windows PowerShell ISE** window, run the following command to start the newly provisioned VMs that will host the Storage Spaces Direct environment:

   ```
   Get-VM -Name 'WSLab-Management' | Start-VM
   Start-Sleep 150
   Get-VM | Where-Object Name -like 'WSLab-S2D*' | Start-VM -AsJob
   ```

5. On the lab VM, start **Hyper-V Manager** and connect via a console session to **WSLab-DC**. When prompted to sign in, provide the username **CORP\LabAdmin** and the password **LS1setup!**.

6. In the **WSLab-DC** VM console session, start **Windows PowerShell ISE** as an administrator.

7. From the **Administrator: Windows PowerShell ISE** window, run `slmgr -rearm` and then select **OK**.

8. From the **Administrator: Windows PowerShell ISE** window, run `Restart-Computer -Force`.

   Note: Make sure that the **WSLab-DC** VM is running before you proceed to the next task.

6.5.3 Task 2: Configure the management server

1. On the lab VM, from **Hyper-V Manager**, and connect with a console session to **WSLab-Management**. When prompted to sign in, provide the username **CORP\LabAdmin** and the password **LS1setup!**.

2. In the **WSLab-Management** VM console session, start **Windows PowerShell ISE** as **Administrator**.

3. In the **WSLab-Management** VM console session, in the **Administrator: Windows PowerShell ISE** window, run the following command to install RSAT:

   ```
   Install-WindowsFeature -Name RSAT-Clustering,RSAT-Clustering-Mgmt,RSAT-Clustering-PowerShell,RSAT-
   ```

   Note: Proceed to the next step without waiting for the installation to complete.

4. In the **WSLab-Management** VM console session, start another instance of **Windows PowerShell ISE** as **Administrator**.

5. From the newly started **Administrator: Windows PowerShell ISE** window, run the following command to download and install the Microsoft Edge (Chromium) browser:

   ```
   $progressPreference='SilentlyContinue'
   Invoke-WebRequest -Uri "https://go.microsoft.com/fwlink/?linkid=2069324&language=en-us&Consent=1"
   Start-Process -FilePath "$env:USERPROFILE\Downloads\MicrosoftEdgeSetup.exe" -Wait
   ```

Note: Proceed to the next step without waiting for the installation to complete.

6. In the **WSLab-Management** VM console session, start another instance of **Windows PowerShell ISE** as **Administrator**.

7. From the newly started **Administrator: Windows PowerShell ISE** window, run the following command to download and install Windows Admin Center:

```
Invoke-WebRequest -UseBasicParsing -Uri https://aka.ms/WACDownload -OutFile "$env:USERPROFILE\Downl
Start-Process msiexec.exe -Wait -ArgumentList "/i $env:USERPROFILE\Downloads\WindowsAdminCenter.msi
```

Note: Proceed to the next step without waiting for the installation to complete.

8. Switch back to the first **Administrator: Windows PowerShell ISE** window where you initiated the installation of RSAT, and wait for the installation to complete.

9. To configure Kerberos-constrained delegation so as to minimize prompts for credentials when using Windows Admin Center, from the **script** pane, run the following command:

```
$gateway = "Management"
$nodes = Get-ADComputer -Filter * -SearchBase "ou=workshop,DC=corp,dc=contoso,DC=com"
$gatewayObject = Get-ADComputer -Identity $gateway
foreach ($node in $nodes){
 Set-ADComputer -Identity $node -PrincipalsAllowedToDelegateToAccount $gatewayObject
}
```

Note: Before you proceed to the next step, verify that the Microsoft Edge and Windows Admin Center installations completed.

10. Close the other two instances of the **Administrator: Windows PowerShell ISE** window you opened earlier in this task.

11. Switch to the **Microsoft Edge** browser window, navigate to `https://management.corp.contoso.com`, and when prompted to authenticate, sign in as **CORP\LabAdmin** with the password **LS1setup!**.

6.5.4 Task 3: Create and configure a failover cluster

1. To provision a failover cluster, In the **WSLab-Management** VM console session, in the **Administrator: Windows PowerShell ISE** window, run the following command:

```
$servers = 1..6 | % {"S2D$_"}
$clusterName = "S2D-Cluster"
$clusterIP = "10.0.0.111"

# Install features on servers
Invoke-Command -computername $servers -ScriptBlock {
    Install-WindowsFeature -Name "Failover-Clustering","Hyper-V-PowerShell","RSAT-Clustering-PowerS
}

# Restart servers since failover clustering in Windows Server 2019 requires reboot
Restart-Computer -ComputerName $servers -Protocol WSMan -Wait -For PowerShell

# Create cluster
New-Cluster -Name $clusterName -Node $servers -StaticAddress $clusterIP
Start-Sleep 5
Clear-DNSClientCache

# Add File Share Witness
# Create a new directory
$witnessName = $clusterName+"Witness"
Invoke-Command -ComputerName DC -ScriptBlock {New-Item -Path c:\Shares -Name $using:WitnessName -It
$accounts = @()
$accounts += "CORP\$($clusterName)$"
$accounts += "CORP\Domain Admins"
New-SmbShare -Name $witnessName -Path "c:\Shares\$witnessName" -FullAccess $accounts -CimSession DC
# Set NTFS permissions
```

```
Invoke-Command -ComputerName DC -ScriptBlock {(Get-SmbShare $using:witnessName).PresetPathAcl | Se
# Set Quorum
Set-ClusterQuorum -Cluster $clusterName -FileShareWitness "\\DC\$witnessName"
```

> **Note:** Wait until the script completes before you proceed to the next task. This should take about 10 minutes.

6.5.5 Task 4: Configure fault domains on the failover cluster

1. To configure fault domains on the Storage Spaces Direct cluster, in the **Administrator: Windows PowerShell ISE** window, run the following command:

```
$clusterName="S2D-Cluster"

# Create Fault domains with PowerShell (note: Enable-ClusterS2D will fail in Windows Server 2016.
New-ClusterFaultDomain -Name "Rack01" -FaultDomainType Rack -Location "Contoso HQ, Room 4010, Aisl
New-ClusterFaultDomain -Name "Rack02" -FaultDomainType Rack -Location "Contoso HQ, Room 4010, Aisl
New-ClusterFaultDomain -Name "Rack03" -FaultDomainType Rack -Location "Contoso HQ, Room 4010, Aisl

# Assign fault domains
# Assign nodes to racks
1..2 |ForEach-Object {Set-ClusterFaultDomain -Name "S2D$_" -Parent "Rack01" -CimSession $clusterNa
3..4 |ForEach-Object {Set-ClusterFaultDomain -Name "S2D$_" -Parent "Rack02" -CimSession $clusterNa
5..6 |ForEach-Object {Set-ClusterFaultDomain -Name "S2D$_" -Parent "Rack03" -CimSession $clusterNa
```

2. To display the newly configured fault domains, in the **Administrator: Windows PowerShell ISE** window, run the following command:

```
$clusterName="S2D-Cluster"
Get-ClusterFaultDomain -CimSession $clusterName
Get-ClusterFaultDomainxml -CimSession $clusterName
```

3. within the Windows Admin Center browser window, connect to the `s2d-cluster.corp.contoso.com` cluster.

4. Review the rack information for each node of the `s2d-cluster.corp.contoso.com` cluster.

> **Note:** If necessary, select **Install** to install **RSAT-Clustering-PowerShell**, which is required by Windows Admin Center.

6.5.6 Task 5: Enable Storage Spaces Direct on the failover cluster

1. To enable Storage Spaces Direct on the cluster, In the **WSLab-Management** VM console session, in the **Administrator: Windows PowerShell ISE** window, run the following command:

```
$clusterName="S2D-Cluster"
Enable-ClusterS2D -CimSession $clusterName -Verbose
```

> **Note:** Wait for the installation to complete before you proceed to the next task. This should take about 3 minutes.

2. Review the provisioning steps and note that the Storage Spaces Direct cluster setup has automatically set the default fault domain awareness on the clustered storage subsystem.

6.5.7 Task 6: Review the fault domain configuration on the Storage Spaces Direct cluster

1. To list storage pool properties, In the **WSLab-Management** VM console session, in the **Administrator: Windows PowerShell ISE** window, run the following command:

```
$clusterName="S2D-Cluster"
Get-StoragePool -CimSession $clusterName -FriendlyName S2D* | fl *
```

> **Note:** Verify that **FaultDomainAwarenessDefault** is automatically set to **StorageRack**.

2. To list properties of storage tiers, in the **Administrator: Windows PowerShell ISE** window, run the following command:

```
Get-StorageTier -CimSession s2d-cluster | fl *
```

> **Note:** Verify that the two tiers named **MirrorOnHDD** and **Capacity** both have **FaultDomainAwarenessDefault** set to **StorageRack**.

6.5.8 Task 7: Create tiered volumes on a Storage Spaces Direct failover cluster

1. To create a volume referencing the **Capacity** tier, In the **WSLab-Management** VM console session, in the **Administrator: Windows PowerShell ISE** window, run the following command:

   ```
   New-Volume -StoragePoolFriendlyName s2d* -FriendlyName WithTier -FileSystem CSVFS_ReFS -StorageTie
   ```

2. To create a volume not referencing any specific tier, in the **Administrator: Windows PowerShell ISE** window, run the following command:

   ```
   New-Volume -StoragePoolFriendlyName s2d* -FriendlyName WithoutTier -FileSystem CSVFS_ReFS -Size 1T
   ```

3. In the **WSLab-Management** VM console session, refresh the Windows Admin Center browser window and navigate to the pane displaying the **WithoutTier** volume properties.
4. On the **WithoutTier** volume pane, note that **Fault domain awareness** is set to **Rack**.

   > **Note:** The actual **FaultDomainAwareness** property is defined on the virtual disk level.

5. Within the Windows Admin Center browser window, navigate to the pane displaying the **WithTier** volume properties.
6. On the **WithTier** volume pane, note that **Fault domain awareness** is also set to **Rack**.

   > **Note:** The actual **FaultDomainAwareness** property is defined on the storage tier associated with the tiered disk.

6.5.9 Task 8: Test resiliency of the Storage Spaces Direct cluster

1. To simulate a failure of an entire rack containing two cluster nodes, switch to the lab VM and in the **Administrator: Windows PowerShell ISE** window, run the following command:

   ```
   Get-VM -Name "WSLab-S2D1","WSLab-S2D2" | Stop-VM -TurnOff
   ```

2. Switch to the console session connected to the **WSLab-Management** VM, and in the browser window displaying Windows Admin Center, navigate to the server inventory of the `s2d-cluster.corp.contoso.com` cluster.
3. Verify that the **S2D1** and **S2D2** nodes in **Rack01** are down, but that the cluster remains online.

   > **Note:** You might need to refresh the page displaying the Windows Admin Center interface to review the updated status of the cluster nodes.

4. Navigate to the volume inventory of the `s2d-cluster.corp.contoso.com` cluster, and then verify that all volumes are healthy.
5. To simulate a storage failure caused by removing one capacity disk from both the **S2D3** and **S2D4** cluster nodes, switch to the lab VM, and in the **Administrator: Windows PowerShell ISE** window, run the following command:

   ```
   Get-VM -Name "WSLab-S2D3","WSLab-S2D4" | Get-VMHardDiskDrive | Where-Object controllerlocation -eq
   ```

6. Switch to the console session connected to the **WSLab-Management** VM, and in the browser window displaying Windows Admin Center, review the volume inventory of the `s2d-cluster.corp.contoso.com` cluster, and verify that all volumes are still healthy.
7. In the **WSLab-Management** VM console session, in the browser window displaying Windows Admin Center, review the summary of drives, and verify that **26** out of **72** drives are listed as **Critical**.
8. To simulate a failure of all disks attached to both **S2D3** and **S2D4** cluster nodes, switch to the lab VM, and in the **Administrator: Windows PowerShell ISE** window, run the following command:

   ```
   Get-VM -Name "WSLab-S2D3","WSLab-S2D4" | Get-VMHardDiskDrive | Where-Object controllerlocation -ne
   ```

9. Switch to the console session connected to the **WSLab-Management** VM, and in the browser window displaying Windows Admin Center, review the summary of drives, and verify that **48** out of **72** drives are listed as **Critical**.

10. Navigate to the volume inventory of the `s2d-cluster.corp.contoso.com` cluster, and verify that all volumes are healthy.

    > **Note:** The storage pool with all volumes might transition to the offline state if you trigger subsequent faults too quickly. If this happens, consider repeating this exercise and pausing between the steps of this task.

    > **Note:** The cluster, pool, and virtual disks are all capable of surviving another node failure providing their respective resources reside in the surviving rack.

11. In the **WSLab-Management** VM console session, start **Failover Cluster Manager** and connect to the `s2d-cluster.corp.contoso.com` cluster.
12. In **Failover Cluster Manager**, identify the owner node of both the **With Tier** and **Without Tier** virtual disks. If the owner node is listed as **S2D3** or **S2D4**, move it to either **S2D5** or **S2D6**.
13. Identify the owner node of the **Cluster Pool 1** storage pool. If the owner node is listed as **S2D3** or **S2D4**, move it to either **S2D5** or **S2D6**.
14. To simulate a **S2D3** node failure, switch to the lab VM, and in the **console** pane of the **Administrator: Windows PowerShell ISE** window, run the following command:

    ```
    Stop-VM -Name "WSLab-S2D3" -TurnOff
    ```

15. Switch to the console session connected to the **WSLab-Management** VM, and then within the console session, in **Failover Cluster Manager**, verify that the **S2D1**, **S2D2**, and **S2D3** nodes are down but that the cluster remains online.
16. In **Failover Cluster Manager**, verify that all virtual disks are online.

    > **Note:** The disks might transition to the offline state if you trigger subsequent faults too quickly.

17. In the **WSLab-Management** VM console session, switch to the browser window displaying Windows Admin Center and verify that all volumes of the `s2d-cluster.corp.contoso.com` cluster are healthy.
18. In the browser window displaying Windows Admin Center, navigate to the `s2d-cluster.corp.contoso.com` cluster's drive inventory, and verify that **48** out of **72** drives are listed as **Critical**.
19. Review the results and verify that the cluster and its virtual disks remain online with only **24** out of **72** physical disks.

    Note: The storage pool with all volumes might transition to the offline state if you trigger subsequent faults too quickly. If this happens, consider repeating this exercise and pausing between the steps of this task.

6.5.10 Task 9: Restore failed disks and nodes on the Storage Spaces Direct cluster

1. To return all disks to the **S2D3** and **S2D4** cluster nodes, switch to the lab VM, and in the **Administrator: Windows PowerShell ISE** window, run the following command:

    ```
    $vmNames="WSLab-S2D3","WSLab-S2D4"
    foreach ($vmName in $vmNames){
      $vhds = (Get-ChildItem -Path "$((get-vm $vmName).ConfigurationLocation)\Virtual Hard Disks" | Whe
      foreach ($vhd in $vhds){
         Add-VMHardDiskDrive -VMName $VMName -Path $VHD
      }
    }
    ```

2. On the lab VM, in the **Administrator: Windows PowerShell ISE** window, run the following command to start the **S2D1**, **S2D2**, and **S2D3** cluster nodes:

    ```
    Start-VM -Name "WSLab-S2D1","WSLab-S2D2","WSLab-S2D3"
    ```

3. Switch to the console session connected to the **WSLab-Management** VM, and in the **Administrator: Windows PowerShell ISE** window, run the following command to identify the repair and regeneration jobs in progress:

    ```
    Get-StorageSubSystem -CimSession s2d-cluster -FriendlyName CL* | Get-StorageJob
    ```

4. In the **WSLab-Management** VM console session, switch to the browser window displaying Windows Admin Center, and then review the status of the drives of the `s2d-cluster.corp.contoso.com` cluster.

5. Within the Windows Admin Center browser window, navigate to the **Dashboard** page of the `s2d-cluster.corp.contoso.com` cluster, and then review its contents to verify that it reports a status of **Healthy** for all cluster components, and that there are no alerts listed.

   > **Note:** The storage subsystem should return to the **Healthy** status in about 5 minutes. So, if you are still getting messages indicating storage faults, you might need to wait and rerun the cmdlet.

6. To identify the health status of the Storage Spaces Direct cluster, In the **WSLab-Management** VM console session, switch to the **Administrator: Windows PowerShell ISE** window and run the following command:

   ```
   Get-HealthFault -CimSession s2d-cluster
   ```

   > **Note:** Ignore faults regarding memory consumption on cluster nodes; these are expected.

6.5.11 Task 10: Deprovision the lab resources

1. Switch to the lab VM.
2. On the lab VM, in the **Administrator: Windows PowerShell ISE** window, run the **F:\WSLab-master\Scripts\Cleanup.ps1** script to remove all VMs provisioned in this lab.
3. In the **Administrator: Windows PowerShell ISE** window, close the tab displaying the **F:\WSLab-master\Scripts\Cleanup.ps1** script.

6.6 Results

6.7 After completing this lab, you will have managed and monitored resiliency of a Storage Spaces Direct cluster. --- lab: title: 'Lab D: Managing Storage Spaces Direct cluster tiers' module: 'Module 3: Planning for and implementing Azure Stack HCI Storage'

7 Lab D: Managing Storage Spaces Direct cluster tiers

7.1 Scenario

Now that you know more about cluster resiliency, you want to explore additional provisions that could help you optimize storage capacity and performance. To accomplish this, you will configure and evaluate storage tiers, including tiers that involve nested resiliency.

7.2 Objectives

After completing this lab, you'll be able to manage Storage Spaces Direct cluster tiers.

7.3 Estimated time: 70 minutes

7.4 Lab setup

To connect to the VM for the lab, follow the steps provided to you by the lab hosting provider.

7.5 Exercise 1: Managing Storage Spaces Direct cluster tiers

7.5.1 Scenario

You want to explore additional provisions that could help you optimize storage capacity and performance by exploring storage tiers, including configuring nested resiliency.

The main tasks for this exercise are as follows:

1. Provision the lab environment VMs.
2. Deploy Storage Spaces Direct clusters.
3. Configure the management server.
4. Configure Storage Spaces Direct cluster tiers.
5. Provision nested tier volumes.
6. Deprovision the lab resources.

7.5.2 Task 1: Provision the lab environment VMs

1. From the lab VM and, in the **Administrator: Windows PowerShell ISE** window, run the following command to rename **LabConfig.ps1** and **Scenario.ps1**:

```
Set-Location -Path 'F:\WSLab-master\Scripts'
Move-Item -Path '.\LabConfig.ps1' -Destination '.\LabConfig.m315.ps1' -Force -ErrorAction Silently
Move-Item -Path '.\Scenario.ps1' -Destination '.\Scenario.m315.ps1' -Force -ErrorAction SilentlyCo
```

2. In the **Administrator: Windows PowerShell ISE** window, from the **script** pane, save the following command as **F:\WSLab-master\Scripts\LabConfig.ps1**:

```
$LabConfig=@{ DomainAdminName='LabAdmin'; AdminPassword='LS1setup!'; Prefix = 'WSLab-'; SwitchName
1..2 | % {$VMNames="2T2node"; $LABConfig.VMs += @{ VMName = "$VMNames$_" ; Configuration = 'S2D' ;
1..3 | % {$VMNames="2T3node"; $LABConfig.VMs += @{ VMName = "$VMNames$_" ; Configuration = 'S2D' ;
1..4 | % {$VMNames="2T4node"; $LABConfig.VMs += @{ VMName = "$VMNames$_" ; Configuration = 'S2D' ;
1..2 | % {$VMNames="3T2node"; $LABConfig.VMs += @{ VMName = "$VMNames$_" ; Configuration = 'S2D' ;
1..3 | % {$VMNames="3T3node"; $LABConfig.VMs += @{ VMName = "$VMNames$_" ; Configuration = 'S2D' ;
1..4 | % {$VMNames="3T4node"; $LABConfig.VMs += @{ VMName = "$VMNames$_" ; Configuration = 'S2D' ;

$LABConfig.VMs += @{
     VMName = "Management" ;
     Configuration = 'S2D' ;
     ParentVHD = 'Win2019_G2.vhdx';
     SSDNumber = 1;
     SSDSize=50GB ;
     MemoryStartupBytes= 4GB;
     NestedVirt=$false;
     StaticMemory=$false;
     VMProcessorCount = 2
 }
```

3. In the **Administrator: Windows PowerShell ISE** window, open and run the **F:\WSLab-master\Scripts\3_Deploy.ps1** script to provision VMs for the Storage Spaces Direct environment.

    Note: Select **None** at the Telemetry prompt. The script should complete in about 15 minutes.

4. When the script completes, in the **Administrator: Windows PowerShell ISE** window, run the following command to start the newly provisioned VMs that will host the Storage Spaces Direct environment:

```
Get-VM -Name 'WSLab-Management' | Start-VM
Start-Sleep 150
Get-VM | Where-Object Name -like 'WSLab-*node*' | Start-VM -AsJob
```

5. On the lab VM, start **Hyper-V Manager** and connect via a console session to **WSLab-DC**. When prompted to sign in, provide the username **CORP\LabAdmin** and the password **LS1setup!**.

6. In the **WSLab-DC** VM console session, start **Windows PowerShell ISE** as an administrator.

7. From the **Administrator: Windows PowerShell ISE** window, run `slmgr -rearm` and then select **OK**.

8. From the **Administrator: Windows PowerShell ISE** window, run `Restart-Computer -Force`.

    Note: Make sure that the **WSLab-DC VM** is running before you proceed to the next task.

7.5.3 Task 2: Configure the management server

1. On the lab VM, from **Hyper-V Manager**, connect via a console session to **WSLab-Management**. When prompted to sign in, provide the **CORP\LabAdmin** username and the password **LS1setup!**.

2. In the **WSLab-Management** VM console session, start **Windows PowerShell ISE** as **Administrator**.

3. In the **Administrator: Windows PowerShell ISE** window, run the following command to install RSAT:

```
Install-WindowsFeature -Name RSAT-Clustering,RSAT-Clustering-Mgmt,RSAT-Clustering-PowerShell,RSAT-
```

    Note: Proceed to the next step without waiting for the installation to complete.

4. In the **WSLab-Management** VM console session, start another instance of **Windows PowerShell ISE** as **Administrator**.

5. From the newly started **Administrator: Windows PowerShell ISE** window, run the following command to download and install Windows Admin Center:

```
Invoke-WebRequest -UseBasicParsing -Uri https://aka.ms/WACDownload -OutFile "$env:USERPROFILE\Downl
Start-Process msiexec.exe -Wait -ArgumentList "/i $env:USERPROFILE\Downloads\WindowsAdminCenter.ms
```

   Note: Proceed to the next step without waiting for the installation to complete.

6. In the **WSLab-Management** VM console session, start another instance of **Windows PowerShell ISE** as **Administrator**.

7. From the newly started **Administrator: Windows PowerShell ISE** window, run the following command to download and install the Microsoft Edge (Chromium) browser:

```
$progressPreference='SilentlyContinue'
Invoke-WebRequest -Uri "https://go.microsoft.com/fwlink/?linkid=2069324&language=en-us&Consent=1" -
Start-Process -FilePath "$env:USERPROFILE\Downloads\MicrosoftEdgeSetup.exe" -Wait
```

   Note: Proceed to the next step without waiting for the installation to complete.

8. To configure Kerberos-constrained delegation to minimize prompts for credentials when using Windows Admin Center, switch back to the first **Administrator: Windows PowerShell ISE** window where you initiated installation of RSAT.

9. Wait for the installation to complete, and then from the **script** pane, run the following command:

```
$gateway = "Management"
$nodes = Get-ADComputer -Filter * -SearchBase "ou=workshop,DC=corp,dc=contoso,DC=com"
$gatewayObject = Get-ADComputer -Identity $gateway
foreach ($node in $nodes){
 Set-ADComputer -Identity $node -PrincipalsAllowedToDelegateToAccount $gatewayObject
}
```

   Note: Before you proceed to the next step, verify that the Microsoft Edge and Windows Admin Center installations completed.

10. Close the other two instances of the **Administrator: Windows PowerShell ISE** window you opened earlier in this task.

7.5.4 Task 3: Deploy Storage Spaces Direct clusters

1. In the **WSLab-Management** VM console session, in the **Administrator: Windows PowerShell ISE** window, run the following command to provision Storage Spaces Direct clusters:

```
$clusters=@()
$clusters+=@{Nodes=1..2 | % {"2T2node$_"} ; Name="2T2nodeClus" ; IP="10.0.0.112" }
$clusters+=@{Nodes=1..3 | % {"2T3node$_"} ; Name="2T3nodeClus" ; IP="10.0.0.113" }
$clusters+=@{Nodes=1..2 | % {"3T2node$_"} ; Name="3T2nodeClus" ; IP="10.0.0.115" }
$clusters+=@{Nodes=1..3 | % {"3T3node$_"} ; Name="3T3nodeClus" ; IP="10.0.0.116" }

# Install features on servers
Invoke-Command -computername $clusters.nodes -ScriptBlock {
  Install-WindowsFeature -Name "Failover-Clustering","Hyper-V-PowerShell","RSAT-Clustering-PowerSh
}

# Restart servers since failover clustering in Windows Server 2019 requires reboot
Restart-Computer -ComputerName $clusters.nodes -Protocol WSMan -Wait -For PowerShell

# Create clusters
foreach ($cluster in $clusters){
  New-Cluster -Name $cluster.Name -Node $cluster.Nodes -StaticAddress $cluster.IP
  Start-Sleep 5
  Clear-DNSClientCache
}
```

```
# Add file share witness
foreach ($cluster in $clusters){
  $clusterName = $cluster.Name
  # Create new directory
  $WitnessName = $clusterName+"Witness"
  Invoke-Command -ComputerName DC -ScriptBlock {New-Item -Path c:\Shares -Name $using:WitnessName -
  $accounts = @()
  $accounts += "CORP\$($clusterName)$"
  $accounts += "CORP\Domain Admins"
  New-SmbShare -Name $WitnessName -Path "c:\Shares\$WitnessName" -FullAccess $accounts -CimSession
  # Set NTFS permissions
  Invoke-Command -ComputerName DC -ScriptBlock {(Get-SmbShare $using:WitnessName).PresetPathAcl | S
  # Set Quorum
  Set-ClusterQuorum -Cluster $clusterName -FileShareWitness "\\DC\$WitnessName"
}

# Enable Storage Spaces Direct and configure mediatype to simulate 3 tier system with SCM (all 800
foreach ($cluster in $clusters.Name){
  Enable-ClusterS2D -CimSession $cluster -Verbose -Confirm:0
  if ($cluster -like "3T*"){
    invoke-command -computername $cluster -scriptblock {
      Get-PhysicalDisk | Where-Object size -eq 800GB | Set-PhysicalDisk -MediaType SCM
      Get-PhysicalDisk | Where-Object size -eq 4TB | Set-PhysicalDisk -MediaType SSD
    }
  }
}
```

Note: Wait for the script to complete before you proceed to the next task. The script should take about 10 minutes to complete. Disregard any errors or warnings.

2. To generate a text file, you will use to import the list of Storage Spaces Direct clusters into Windows Admin Center, In the **WSLab-Management** VM console session, from the **console** pane of the **Administrator: Windows PowerShell ISE** window, run the following command:

```
(Get-Cluster -Domain $env:userdomain | Where-Object S2DEnabled -eq 1).Name | Out-File c:\s2dcluste
```

3. Switch to the Microsoft Edge browser window, select **Get started**, accept the default tab page settings, and then select the **Continue without Signing-in** link.

4. Use the Microsoft Edge browser to navigate to `https://management.corp.contoso.com`, and then when prompted to authenticate, sign in as **CORP\LabAdmin** with the password **LS1setup!**.

5. In the **WSLab-Management** VM console session, within the browser window displaying the Windows Admin Center interface, connect to all Storage Spaces Direct clusters you deployed earlier in this exercise by importing their names from the **c:\s2dclusters.txt** file.

7.5.5 Task 4: Configure Storage Spaces Direct cluster tiers

1. In the **WSLab-Management** VM console session, in the **Administrator: Windows PowerShell ISE** window, run the following command to identify the tiers on a two-node cluster with HDDs only:

```
Get-StorageTier -CimSession 2T2NodeClus |
  ft FriendlyName,MediaType,ResiliencySettingName,NumberOfDataCopies,PhysicalDiskRedundancy,FaultDo
```

Note: On a two-node Storage Spaces Direct cluster with HDDs only, there are 2 tiers: one is Capacity, created to provide compatibility with Windows Server 2016 naming convention; the other is MirrorOnHDD, which follows the new naming convention in Windows Server 2019. The value of **NumberOfDataCopies** represents the **2way mirror** configuration, and **PhysicalDiskRedundancy** reflects the ability to tolerate a single fault. The value of **FaultDomainAwareness** indicates that the two copies are distributed across instances of **StorageScaleUnit**. The number of columns is automatically calculated, depending on the number of nodes and disks in each node, and is assigned during creation of virtual disks. The value of **NumberOfGroups** indicates the parity setting, which in this case, is set to **1**.

2. In the **WSLab-Management** VM console session, in the **Administrator: Windows PowerShell ISE** window, run the following command to identify all of the tiers on all of the Storage Spaces Direct clusters in the lab environment:

```
$clusters=(Get-Cluster -Domain $env:userdomain | Where-Object S2DEnabled -eq 1).Name
Get-StorageTier -CimSession $clusters |
  Sort-Object PSComputerName |
  ft PSComputerName,FriendlyName,MediaType,ResiliencySettingName,NumberOfDataCopies,PhysicalDiskRe
```

> **Note:** The tiers are generated automatically when you invoke the **Enable-ClusterS2D** PowerShell cmdlet. Tiers reflect the media type present in cluster nodes.

3. From **Administrator: Windows PowerShell ISE** window, run the following command to identify the Windows Server 2019-specific mirror tiers on all the Storage Spaces Direct clusters in the lab environment. (These are tiers that reference **MirrorOnHDD**, **MirrorOnSSD**, and **MirrorOnSCM**, where *SCM* designates Storage Class Memory):

```
$clusters=(Get-Cluster -Domain $env:userdomain | Where-Object S2DEnabled -eq 1).Name
Get-StorageTier -CimSession $clusters |
  Where-Object friendlyname -like mirror* |
  Sort-Object PSComputerName |
  ft PSComputerName,FriendlyName,MediaType,ResiliencySettingName,NumberOfDataCopies,PhysicalDiskRe
```

> **Note:** The values of **NumberOfCopies** and **PhysicalDiskRedundancy** is **2** on two-node clusters, and **3** for clusters with three or more nodes.

4. In the **Administrator: Windows PowerShell ISE** window, run the following command to identify the Windows Server 2019-specific parity tiers on all of the Storage Spaces Direct clusters in the lab environment. (These are the tiers that reference **ParityOnHDD**, **ParityOnSSD**, and **ParityOnSCM**, where *SCM* designates Storage Class Memory*):

```
$clusters=(Get-Cluster -Domain $env:userdomain | Where-Object S2DEnabled -eq 1).Name
Get-StorageTier -CimSession $clusters |
  Where-Object friendlyname -like parity* |
  Sort-Object PSComputerName |
  ft PSComputerName,FriendlyName,MediaType,ResiliencySettingName,NumberOfDataCopies,PhysicalDiskRe
```

> **Note:** The script doesn't return any results because by default, parity tiers are created only on Storage Spaces Direct clusters with four or more nodes. With Windows Server 2019-based Storage Spaces Direct clusters, you have the option to create parity tiers by implementing nested resiliency on two-node clusters.

5. In the **Administrator: Windows PowerShell ISE** window, run the following command to create nested resiliency tiers:

```
#Select clusters to fix tiers
$clusterNames=(Get-Cluster -Domain $env:userdomain | Where-Object S2DEnabled -eq 1 | Out-GridView

foreach ($clusterName in $clusterNames){
   $storageTiers=Get-StorageTier -CimSession $clusterName
   $numberOfNodes=(Get-ClusterNode -Cluster $clusterName).Count
   $MediaTypes=(Get-PhysicalDisk -CimSession $clusterName |where mediatype -ne Unspecified | Where-
   $clusterFunctionalLevel=(Get-Cluster -Name $clusterName).ClusterFunctionalLevel

   foreach ($mediaType in $mediaTypes){
      if ($numberOfNodes -eq 2) {
         # Create Mirror Tiers
         if (-not ($storageTiers | Where-Object FriendlyName -eq "MirrorOn$mediaType")){
            New-StorageTier -CimSession $clusterName -StoragePoolFriendlyName "S2D on $clusterName
         }
         if ($clusterFunctionalLevel -ge 10){
            # Create NestedMirror Tiers
            if (-not ($storageTiers | Where-Object FriendlyName -eq "NestedMirrorOn$mediaType")){
               New-StorageTier -CimSession $clusterName -StoragePoolFriendlyName "S2D on $clusterNa
            }
            #Create NestedParity Tiers
            if (-not ($storageTiers | Where-Object FriendlyName -eq "NestedParityOn$mediaType")){
               New-StorageTier -CimSession $clusterName -StoragePoolFriendlyName "S2D on $clusterNa
            }
         }
      } elseif ($numberOfNodes -eq 3) {
         #Create Mirror Tiers
         if (-not ($storageTiers | Where-Object FriendlyName -eq "MirrorOn$mediaType")){
            New-StorageTier -CimSession $clusterName -StoragePoolFriendlyName "S2D on $clusterName
         }
      } elseif ($numberOfNodes -ge 4) {
         #Create Mirror Tiers
         if (-not ($storageTiers | Where-Object FriendlyName -eq "MirrorOn$mediaType")){
           New-StorageTier -CimSession $clusterName -StoragePoolFriendlyName "S2D on $clusterName"
         }
         #Create Parity Tiers
         if (-not ($storageTiers | Where-Object FriendlyName -eq "ParityOn$mediaType")){
           New-StorageTier -CimSession $clusterName -StoragePoolFriendlyName "S2D on $clusterName"
         }
      }
   }
}
```

6. When prompted, in the **Select Clusters to Check on tiers** window, select both the **2T2nodeClus** and **3T2nodeClus** entries, and then select **OK**.

7. In the **Administrator: Windows PowerShell ISE** window, run the following command to identify the Windows Server 2019-specific nested tiers on two-node Storage Spaces Direct clusters in the lab environment:

```
$clusters=(Get-Cluster -Domain $env:userdomain | Where-Object S2DEnabled -eq 1).Name
Get-StorageTier -CimSession $clusters |
  Where-Object friendlyname -like nested* |
  Sort-Object PSComputerName |
  ft PSComputerName,FriendlyName,MediaType,ResiliencySettingName,NumberOfDataCopies,PhysicalDiskRe
```

Note: By default, parity tiers are created only on Storage Spaces Direct clusters with four or more nodes. With Windows Server 2019-based Storage Spaces Direct clusters, you have the option to create parity tiers by implementing nested resiliency on two-node clusters.

7.5.6 Task 5: Provision nested tier volumes

1. In the **Administrator: Windows PowerShell ISE** window, run the following commands to provision a volume on the Storage Spaces Direct cluster **2T2nodeClus**, based on the **NestedMirrorOnHDD** nested resiliency tier:

```
$clusterName = '2T2nodeClus'
New-Volume -StoragePoolFriendlyName s2d* -FriendlyName NestedMirroronHDDVolume -FileSystem CSVFS_Re
```

2. In the **WSLab-Management** VM console session, within the Windows Admin Center browser window, connect to the **2T2nodeClus** cluster.

3. In the Windows Admin Center, navigate to the volume inventory of the **2T2nodeClus** cluster.

4. In the list of volumes, note that the **NestedMirroronHDDVolume** has resiliency set to **Nested two-way mirror**.

5. Within the Windows Admin Center, navigate back to drive inventory of the **2T2nodeClus** cluster.

6. In the list of drives, review the list of drive types.

7. In the **WSLab-Management** VM console session, in the **Administrator: Windows PowerShell ISE** window, run the following command to provision a volume on the Storage Spaces Direct cluster **3T2nodeClus**, based on the **NestedMirrorOnSSD** nested resiliency tier:

```
$clusterName = '3T2nodeClus'
New-Volume -StoragePoolFriendlyName s2d* -FriendlyName NestedMirroronSSDVolume -FileSystem CSVFS_Re
```

8. In the **WSLab-Management** VM console session, within the Windows Admin Center browser window, connect to the **3T2nodeClus** cluster.
9. In the Windows Admin Center, navigate to the volume inventory of the **3T2nodeClus** cluster.
10. In the list of volumes, note that the **NestedMirroronSSDVolume** has resiliency set to **Nested two-way mirror**.
11. In the Windows Admin Center, navigate back to the drive inventory of the **3T2nodeClus** cluster.
12. In the list of drives, review the list of drive types.

7.5.7 Task 6: Deprovision the lab resources

1. Switch to the lab VM.
2. On the lab VM, in the **Administrator: Windows PowerShell ISE** window, run the **F:\WSLab-master\Scripts\Cleanup.ps1** script to remove all VMs provisioned in this lab.
3. In the **Administrator: Windows PowerShell ISE** window, close the tab displaying the **F:\WSLab-master\Scripts\Cleanup.ps1** script.

7.6 Results

7.7 After completing this lab, you will have managed Storage Spaces Direct cluster tiers. --- lab: title: 'Lab E: Identifying and analyzing metadata of a Storage Spaces Direct cluster (optional)' module: 'Module 3: Planning for and implementing Azure Stack HCI Storage'

8 Lab E: Identifying and analyzing metadata of a Storage Spaces Direct cluster (optional)

8.1 Scenario

Another resiliency consideration you want to explore is metadata of the storage pool and its components. You want to ensure that you understand resiliency provisions that must be taken into account to protect cluster stability and integrity. You also want to be able to identify how a Storage Spaces Direct cluster maintains information about its data.

8.2 Objectives

After completing this lab, you'll be able to identify and analyze the metadata of a Storage Spaces Direct cluster.

8.3 Estimated time: 35 minutes

8.4 Lab setup

To connect to the VM for the lab, follow the steps provided to you by the lab hosting provider.

8.5 Exercise 1: Identifying and analyzing metadata of a Storage Spaces Direct cluster

8.5.1 Scenario

To ensure that you understand resiliency provisions that must be taken into account to protect cluster stability and integrity, and to be able to identify how the Storage Spaces Direct cluster maintains information about its data, you need to identify and analyze the metadata of the storage pool and its components.

The main tasks for this exercise are as follows:

1. Provision the lab environment VMs.
2. Deploy a Storage Spaces Direct cluster.
3. Examine physical disk owners.
4. Explore storage pool metadata.
5. Explore metadata of a volume.
6. Explore metadata of a scoped volume.

7. Deprovision the lab resources.

8.5.2 Task 1: Provision the lab environment VMs

1. From the lab VM, in the **Administrator: Windows PowerShell ISE** window, run the following command to rename **LabConfig.ps1** and **Scenario.ps1**:

```
Set-Location -Path 'F:\WSLab-master\Scripts'
Move-Item -Path '.\LabConfig.ps1' -Destination '.\LabConfig.m314.ps1' -Force -ErrorAction Silently
Move-Item -Path '.\Scenario.ps1' -Destination '.\Scenario.m314.ps1' -Force -ErrorAction SilentlyCo
```

2. In the **Administrator: Windows PowerShell ISE** window, from the **script** pane, save the following command as **F:\WSLab-master\Scripts\LabConfig.ps1**:

```
$LabConfig=@{ DomainAdminName = 'LabAdmin'; AdminPassword = 'LS1setup!'; Prefix = 'WSLab-'; Switch
1..6 | % {$VMNames = "6node"; $LABConfig.VMs += @{ VMName = "$VMNames$_" ; Configuration = 'S2D' ;
$LABConfig.VMs += @{
     VMName = "Management" ;
     Configuration = 'S2D' ;
     ParentVHD = 'Win2019_G2.vhdx';
     SSDNumber = 1;
     SSDSize = 50GB ;
     MemoryStartupBytes = 8GB;
     NestedVirt = $false;
     StaticMemory = $true;
     VMProcessorCount = 4
}
```

3. On the lab VM, in the **Administrator: Windows PowerShell ISE** window, open and run the **F:\WSLab-master\Scripts\3_Deploy.ps1** script to provision VMs for the Storage Spaces Direct environment.

   > **Note:** Select **None** at the Telemetry prompt. The script should complete in about 15 minutes.

4. When the script completes, in the **Administrator: Windows PowerShell ISE** window, run the following command to start the newly provisioned VMs that will host the Storage Spaces Direct environment:

```
Get-VM -Name 'WSLab-Management' | Start-VM
Start-Sleep 150
Get-VM | Where-Object Name -like 'WSLab-*node*' | Start-VM -AsJob
```

5. On the lab VM, start **Hyper-V Manager** and connect via a console session to **WSLab-DC**. When prompted to sign in, provide the username **CORP\LabAdmin** and the password **LS1setup!**.

6. In the **WSLab-DC** VM console session, start **Windows PowerShell ISE** as an administrator.

7. From the **Administrator: Windows PowerShell ISE** window, run `slmgr -rearm` and then select **OK**.

8. From the **Administrator: Windows PowerShell ISE** window, run `Restart-Computer -Force`.

   Note: Make sure that the **WSLab-DC** VM is running before you proceed to the next task.

8.5.3 Task 2: Deploy a Storage Spaces Direct cluster

1. On the lab VM, from **Hyper-V Manager**, and connect via a console session to **WSLab-Management**. When prompted to sign in, provide the username **CORP\LabAdmin** and the password **LS1setup!**.

2. In the **WSLab-Management** VM console session, start **Windows PowerShell ISE** as **Administrator**.

3. In the **Administrator: Windows PowerShell ISE** window, run the following command to provision a Storage Spaces Direct cluster:

```
$clusters = @()
$clusters += @{Nodes=1..6 | % {"6node$_"} ; Name="6nodeCluster" ; IP="10.0.0.116" }

Install-WindowsFeature -Name RSAT-Clustering,RSAT-Clustering-Mgmt,RSAT-Clustering-PowerShell,RSAT-

# Install features on servers
```

```
Invoke-Command -computername $clusters.nodes -ScriptBlock {
   Install-WindowsFeature -Name "Failover-Clustering","Hyper-V-PowerShell"
}

# Restart all servers to finalize installation of Failover Clustering
Restart-Computer -ComputerName $clusters.nodes -Protocol WSMan -Wait -For PowerShell

# Create clusters
foreach ($cluster in $clusters){
   New-Cluster -Name $cluster.Name -Node $cluster.Nodes -StaticAddress $cluster.IP
   Start-Sleep 5
   Clear-DNSClientCache
}

# Add file share witness
foreach ($cluster in $clusters){
   $clusterName = $cluster.Name
   # Create new directory
   $witnessName = $clusterName+"Witness"
   Invoke-Command -ComputerName DC -ScriptBlock {New-Item -Path c:\Shares -Name $using:witnessName
   $accounts = @()
   $accounts += "CORP\$($clusterName)$"
   $accounts += "CORP\Domain Admins"
   New-SmbShare -Name $witnessName -Path "c:\Shares\$witnessName" -FullAccess $accounts -CimSessio
   # Set NTFS permissions
   Invoke-Command -ComputerName DC -ScriptBlock {(Get-SmbShare $using:witnessName).PresetPathAcl |
   # Set Quorum
   Set-ClusterQuorum -Cluster $clusterName -FileShareWitness "\\DC\$WitnessName"
}

# Enable Storage Spaces Direct
Enable-ClusterS2D -CimSession $clusters.Name -Verbose -Confirm:0
```

Note: Wait for the script to complete. This might take about 5 minutes.

8.5.4 Task 3: Examine physical disk owners

1. In the **WSLab-Management** VM console session, in the **Administrator: Windows PowerShell ISE** window, run the following command to display the physical disks of **6nodecluster** Storage Spaces Direct cluster:

   ```
   Get-PhysicalDisk -CimSession 6nodecluster |ft FriendlyName,Size,Description
   ```

2. In the **Administrator: Windows PowerShell ISE** window, run the following command to set the **Description** attribute of each physical disk to the name of the cluster node to which the disk is attached for all Storage Spaces Direct clusters:

   ```
   $clusters = @()
   $clusters += @{Nodes=1..6 | % {"6node$_"} ; Name="6nodeCluster" ; IP="10.0.0.116" }
   foreach ($clusterName in ($clusters.Name | select -Unique)){
      $storageNodes=Get-StorageSubSystem -CimSession $clusterName -FriendlyName Clus* | Get-StorageNo
      foreach ($storageNode in $storageNodes){$storageNode | Get-PhysicalDisk -PhysicallyConnected -C
   }
   ```

3. In the **Administrator: Windows PowerShell ISE** window, re-run the following command to display the physical disks of the **6nodecluster** Storage Spaces Direct cluster, this time with the **Description** attribute containing the name of the owner node:

   ```
   Get-PhysicalDisk -CimSession 6nodecluster |ft FriendlyName,Size,Description
   ```

8.5.5 Task 4: Explore storage pool metadata

1. In the **WSLab-Management** VM console session, in the **Administrator: Windows PowerShell ISE** window, run the following command to display the number of disks with metadata for the 6-node cluster:

```
foreach ($clusterName in ($clusters.Name | select -Unique)){
  Get-StoragePool -CimSession $clusterName |
  Get-PhysicalDisk -HasMetadata -CimSession $clusterName |
  Sort-Object Description |
  Format-Table DeviceId,FriendlyName,SerialNumber,MediaType,Description
}
```

> **Note:** The number of disks with metadata depends on size of the cluster, as per the following table:

Table 1: Number of disks with metadata

Number of nodes (fault domains)	Number of disks with metadata
2	6
3	6
4	8
5	5
6	5

> **Note:** In a 6-node cluster, metadata is located on five of the six nodes. This means that if you lose random half nodes, 50 percent of the storage pool will go offline.

2. In the **WSLab-Management** VM console session, start **Failover Cluster Manager**, and then connect to `6nodecluster.corp.contoso.com`.
3. In **Failover Cluster Manager**, review the list of nodes.
4. Locate the storage pool named **Cluster Pool 1**, and examine its properties, including virtual disks and physical disks.
5. In the **WSLab-Management** VM console session, switch to the **Administrator: Windows PowerShell ISE** window and run the following command to capture the list of three nodes hosting the storage pool metadata of the 6-node cluster:

```
$nodesToShutDown = (Get-StoragePool -CimSession 6nodecluster |
Get-PhysicalDisk -HasMetadata -CimSession $clusterName | Select-Object -First 3).Description
```

6. In the **Administrator: Windows PowerShell ISE** window, run the following command to shut down the three nodes hosting metadata of the 6-node cluster:

```
Stop-Computer -ComputerName $nodesToShutDown -Force
```

7. In the **WSLab-Management** VM console session, switch to **Failover Cluster Manager** and verify that the **Cluster Pool 1** storage pool has a status of **Failed**.

   > **Note:** It might take a few minutes before the storage pool reaches the **Failed** status.

8. In **Failover Cluster Manager**, select **Show Critical Events** to review the cluster's critical events, and locate the most recent critical event that references the storage pool failure resulting from a lack of quorum of healthy disks.
9. Switch to the lab VM, and then in the Hyper-V Manager console, start the VMs that you shut down earlier in this exercise.
10. Switch to the **WSLab-Management** VM, and in **Failover Cluster Manager**, bring the **Cluster Pool 1** online.

8.5.6 Task 5: Explore metadata of a volume

1. In the **WSLab-Management** VM console session, in the **Administrator: Windows PowerShell ISE** window, run the following command to create a volume on the 6-node cluster:

```
Invoke-Command -ComputerName ($clusters.Name | select -Unique) -ScriptBlock {New-Volume -FriendlyN
```

2. In the **Administrator: Windows PowerShell ISE** window, run the following command to display metadata of the newly created volume on the 6-node cluster:

```
foreach ($clusterName in ($clusters.Name | select -Unique)){
   Get-VirtualDisk -FriendlyName labVolume -CimSession $clusterName |
   Get-PhysicalDisk -HasMetadata -CimSession $clusterName |
   Sort-Object Description |
   Format-table DeviceId,FriendlyName,SerialNumber,MediaType,Description
}
```

> **Note:** Based on the output, you can determine whether the number of disks with metadata matches the one used by the storage pool metadata.

8.5.7 Task 6: Explore metadata of a scoped volume

1. In the **Administrator: Windows PowerShell ISE** window, run the following command to create scoped volumes on the 6-node cluster:

```
$faultDomains = Get-StorageFaultDomain -Type StorageScaleUnit -CimSession 6nodecluster | Sort Frie
New-Volume -FriendlyName "2Scope-Volume" -Size 100GB -StorageFaultDomainsToUse ($faultDomains | Ge
New-Volume -FriendlyName "3Scope-Volume" -Size 100GB -StorageFaultDomainsToUse ($faultDomains | Ge
New-Volume -FriendlyName "4Scope-Volume" -Size 100GB -StorageFaultDomainsToUse ($faultDomains | Ge
New-Volume -FriendlyName "5Scope-Volume" -Size 100GB -StorageFaultDomainsToUse ($faultDomains | Ge
New-Volume -FriendlyName "6Scope-Volume" -Size 100GB -StorageFaultDomainsToUse ($faultDomains | Ge
```

> **Note:** For a 6-node cluster, set the number of a volume's scopes to four. The additional volumes in this exercise aren't used as an example of their practical use, but rather as an illustration about how different scope values affect volume distribution.

2. In the **Administrator: Windows PowerShell ISE** window, run the following command to display metadata of the newly created volumes on the 6-node cluster:

```
$friendlyNames=2..6 | % {"$($_)Scope-Volume"}
foreach ($friendlyName in $friendlyNames){
   Write-Host -Object "$friendlyName" -ForeGroundColor Cyan
   Get-VirtualDisk -FriendlyName $friendlyName -CimSession 6nodecluster |
   Get-PhysicalDisk -HasMetadata -CimSession 6nodecluster |
   Sort-Object Description |
   Format-Table DeviceId,FriendlyName,SerialNumber,MediaType,Description
}
```

> **Note:** Review the output and note that the number of cluster nodes containing metadata of each volume matches the scope determined by the value of the **StorageFaultDomainsToUse** parameter of the **New-Volume** cmdlet.

8.5.8 Task 7: Deprovision the lab resources

1. Switch to the lab VM.
2. On the lab VM, in the **Administrator: Windows PowerShell ISE** window, run the **F:\WSLab-master\Scripts\Cleanup.ps1** script to remove all VMs provisioned in this lab.
3. In the **Administrator: Windows PowerShell ISE** window, close the tab displaying the **F:\WSLab-master\Scripts\Cleanup.ps1** script.

8.6 Results

8.7 After completing this lab, you will have identified and analyzed the metadata of a Storage Spaces Direct cluster.

8.8 lab: title: 'Lab A: Deploying Software-Defined Networking' module: 'Module 4: Planning for and Implementing Azure Stack HCI Networking'

9 Lab A: Deploying Software-Defined Networking

9.1 Scenario

To address the requirements for deploying an isolated VDI farm for users in the Contoso Securities Research department, which is supposed to replace an aging Windows Server 2012 R2–based RDS deployment, you'll implement Software-Defined Networking (SDN) on hyperconverged infrastructure. As the first step in this process, you need to provision the SDN infrastructure by using the scripts available online.

9.2 Objectives

After completing this lab, you'll be able to deploy SDN by using PowerShell.

9.3 Estimated time: 120 minutes

9.4 Lab setup

To connect to the lab VM, follow the steps the lab hosting provider provides you.

9.5 Exercise 1: Deploying Software-Defined Networking by using PowerShell

9.5.1 Scenario

To prepare for the rest of this lab, you need to provision the Software-Defined Networking (SDN) infrastructure by leveraging the scripts available at microsoft/WSLab.

The main tasks for this exercise are as follows:

1. Deploy the VMs that will serve as the SDN infrastructure Hyper-V hosts.
2. Deploy the SDN infrastructure VMs.

9.5.2 Task 1: Deploy the VMs that will serve as the SDN infrastructure Hyper-V hosts

1. On the lab VM, start Windows PowerShell ISE as Administrator and run the following command to remove the **Zone.Identifier** alternate data stream, which has a value of **3** indicating that it was downloaded from the internet:

   ```
   Get-ChildItem -Path F:\WSLab-master\ -File -Recurse | Unblock-File
   ```

2. On the lab VM, from the Administrator: Windows PowerShell ISE window, run the following command to set the current directory:

   ```
   Set-Location -Path F:\WSLab-master\Scripts
   ```

3. On the lab VM, from the Administrator: Windows PowerShell ISE window, run the following command to rename **LabConfig.ps1** and **Scenario.ps1**:

   ```
   Move-Item -Path '.\LabConfig.ps1' -Destination '.\LabConfig.m410.ps1' -Force -ErrorAction Silently
   Move-Item -Path '.\Scenario.ps1' -Destination '.\Scenario.m410.ps1' -Force -ErrorAction SilentlyCo
   ```

4. On the lab VM, in the Administrator: Windows PowerShell ISE window, open a new tab in the **script** pane, paste the following content and save it as **F:\WSLab-master\Scripts\LabConfig.ps1**:

   ```
   $LabConfig=@{ DomainAdminName = 'LabAdmin'; AdminPassword = 'LS1setup!'; Prefix = 'SDNExpress2019-
   $LABConfig.AdditionalNetworksConfig += @{
        NetName = 'HNV';
        NetAddress = '10.103.33.';
        NetVLAN = '201';
        Subnet = '255.255.255.0'
    }

   1..4 | % {
   $VMNames = "HV";
   $LABConfig.VMs += @{
        VMName = "$VMNames$_";
        Configuration = 'S2D';
        ParentVHD = 'Win2019Core_G2.vhdx';
        SSDNumber = 2;
        SSDSize = 800GB;
        HDDNumber = 4;
        HDDSize = 4TB;
        MemoryStartupBytes = 20GB;
        NestedVirt = $True;
   ```

```
     StaticMemory = $True;
     VMProcessorCount = 6
    }
}

$LABConfig.VMs += @{
     VMName = "Management";
     Configuration = 'S2D';
     ParentVHD = 'Win2019_G2.vhdx';
     SSDNumber = 1;
     SSDSize = 50GB;
     MemoryStartupBytes = 4GB;
     NestedVirt = $false;
     StaticMemory = $false;
     VMProcessorCount = 4
}
```

5. Copy the **Scenario.ps1** and **MultiNodeConfig.psd1** files from **F:\WSLab-master\Scenarios\SDNExpress with Windows Admin Center** to **F:\WSLab-master\Scripts**.

6. From the Windows PowerShell ISE window, run the **F:\WSLab-master\Scripts\3_Deploy.ps1** script to provision **SDNExpress2019-DC** based on the **DC** VM and the remaining VMs for the SDN environment.

   Note: Select **None** at the Telemetry prompt. The script should complete in about 7 minutes.

7. In the Windows PowerShell ISE window, open the **F:\WSLab-master\Scripts\Scenario.ps1** script, remove all content following the line **128**, starting from `# ENDING Run from Hyper-V Host ENDING #`, and then save the modified file as **Scenario_Part1.ps1**.

   Note: You must run this part of the scenario script from the Hyper-V host.

8. In the Windows PowerShell ISE window, run the **F:\WSLab-master\Scripts\Scenario_Part1.ps1** script to configure the VMs that will host the lab environment. When prompted for the location of the parent VHDX for the SDN VMs, point to **F:\WSLab-master\Scripts\ParentDisks\Win2019Core_G2.vhdx**. When prompted for the **MultiNodeConfig.psd1** file, point to the file you copied to **F:\WSLab-master\Scripts**. When prompted for Windows Admin Center MSI, point to the downloaded Windows Installer file in the **F:\Source** folder.

   Note: If the script fails with the message **Copy-VMFile : Failed to initiate copying files to the guest**, rerun the script.

   Note: The script should complete in about 15 minutes.

   Note: Ignore the error following the line **ScriptHalted** and message prompting to restart **SDNExpress2019-Management**. That's expected.

9. After the script completes, in the Windows PowerShell ISE window, run the following script to expand the size of the disks hosting drive **C** of the newly provisioned VMs that will host the SDN environment:

```
$servers = @('SDNExpress2019-HV1','SDNExpress2019-HV2','SDNExpress2019-HV3','SDNExpress2019-HV4')
$paths = (Get-VM -Name $servers | Get-VMHardDiskDrive | Where-Object {$_.ControllerLocation -eq 0}
foreach ($path in $paths) { Resize-VHD -Path $path -SizeBytes 100GB }
```

9.5.3 Task 2: Deploy the SDN infrastructure VMs

Note: Sign in to the **DC** VM using the **CORP\LabAdmin** username and **LS1setup!** password, run `slmgr -rearm` and restart it.

1. On the lab VM, use the Hyper-V Manager console to connect to the **SDNExpress2019-Management** VM. When prompted to sign in, provide the **CORP\LabAdmin** username and **LS1setup!** password.

2. Within the console session to the **SDNExpress2019-Management** VM, start Windows PowerShell ISE as Administrator and run the following script to expand the size of drive **C** of the VMs that will host the SDN environment:

```
$servers = @('HV1','HV2','HV3','HV4')
Invoke-Command -ComputerName $servers -ScriptBlock {
  $size = Get-PartitionSupportedSize -DriveLetter C
  Resize-Partition -DriveLetter C -Size $size.SizeMax
}
```

3. On the lab VM, from the **script** pane of the Administrator: Windows PowerShell ISE window, run the following commands to download the following file:

   ```
   New-Item F:\Allfiles -itemtype directory -Force
   Invoke-Webrequest -Uri "https://raw.githubusercontent.com/MicrosoftLearning/WS-013T00-Azure-Stack-
   ```

4. Within the console session to the **SDNExpress2019-Management** VM, start File Explorer and navigate to the **C:\Library** folder.

5. Switch back to the lab VM and use the copy and paste functionality of the **Hyper-V** console session to copy **F:\WSLab-master\Scripts\Scenario.ps1** and **F:\Allfiles\SDNExpressModule.psm1** on the lab VM to **C:\Library** on the **SDNExpress2019-Management** VM.

6. Within the console session to the **SDNExpress2019-Management** VM, in the Administrator: Windows PowerShell ISE window, open the **C:\Library\Scenario.ps1** script, and comment out line 375 so it looks like so: `# Expand-Archive -Path C:\SDN-Master.zip -DestinationPath C:\Library`

7. Within the console session to the **SDNExpress2019-Management** VM, in the Administrator: Windows PowerShell ISE window, remove all content before the line 136, up to the line prior to `# Run from DC / VMM #`, and then save the modified file as **Scenario_Part2.ps1**.

8. Within the console session to the **SDNExpress2019-Management** VM, in the Administrator: Windows PowerShell ISE window, run the following command:

   ```
   Expand-Archive -Path C:\SDN-Master.zip -DestinationPath C:\Library
   Copy-Item -Path C:\Library\SDNExpressModule.psm1 -Destination C:\Library\SDN-master\SDNExpress\scr
   ```

   > **Note**: This part of the scenario script needs to be run from the management VM.

9. Within the console session to the **SDNExpress2019-Management** VM, in the Administrator: Windows PowerShell ISE window, run the newly saved **C:\Library\Scenario_Part2.ps1** script to configure the SDN VMs.

   > **Note**: Wait until the script completes before you proceed. The script should complete in about 90 minutes. Disregard any cluster validation errors.

9.5.4 Results

9.6 After completing this lab, you will have successfully provisioned the SDN infrastructure.

9.7 lab: title: 'Lab A: Deploying Software-Defined Networking' module: 'Module 4: Planning for and Implementing Azure Stack HCI Networking'

10 Lab A: Deploying Software-Defined Networking

10.1 Scenario

To address the requirements for deploying an isolated VDI farm for users in the Contoso Securities Research department, which is supposed to replace an aging Windows Server 2012 R2–based RDS deployment, you'll implement Software-Defined Networking (SDN) on hyperconverged infrastructure. As the first step in this process, you need to provision the SDN infrastructure by using the scripts available online.

10.2 Objectives

After completing this lab, you'll be able to deploy SDN by using PowerShell.

10.3 Estimated time: 120 minutes

10.4 Lab setup

To connect to the lab VM, follow the steps the lab hosting provider provides you.

10.5 Exercise 1: Deploying Software-Defined Networking by using PowerShell

10.5.1 Scenario

To prepare for the rest of this lab, you need to provision the Software-Defined Networking (SDN) infrastructure by leveraging the scripts available at microsoft/WSLab.

The main tasks for this exercise are as follows:

1. Deploy the VMs that will serve as the SDN infrastructure Hyper-V hosts.
2. Deploy the SDN infrastructure VMs.

10.5.2 Task 1: Deploy the VMs that will serve as the SDN infrastructure Hyper-V hosts

1. On the lab VM, start Windows PowerShell ISE as Administrator and run the following command to remove the **Zone.Identifier** alternate data stream, which has a value of **3** indicating that it was downloaded from the internet:

   ```
   Get-ChildItem -Path F:\WSLab-master\ -File -Recurse | Unblock-File
   ```

2. On the lab VM, from the Administrator: Windows PowerShell ISE window, run the following command to set the current directory:

   ```
   Set-Location -Path F:\WSLab-master\Scripts
   ```

3. On the lab VM, from the Administrator: Windows PowerShell ISE window, run the following command to rename **LabConfig.ps1** and **Scenario.ps1**:

   ```
   Move-Item -Path '.\LabConfig.ps1' -Destination '.\LabConfig.m410.ps1' -Force -ErrorAction Silently
   Move-Item -Path '.\Scenario.ps1' -Destination '.\Scenario.m410.ps1' -Force -ErrorAction SilentlyCo
   ```

4. On the lab VM, in the Administrator: Windows PowerShell ISE window, open a new tab in the **script** pane, paste the following content and save it as **F:\WSLab-master\Scripts\LabConfig.ps1**:

   ```
   $LabConfig=@{ DomainAdminName = 'LabAdmin'; AdminPassword = 'LS1setup!'; Prefix = 'SDNExpress2019-
   $LABConfig.AdditionalNetworksConfig += @{
        NetName = 'HNV';
        NetAddress = '10.103.33.';
        NetVLAN = '201';
        Subnet = '255.255.255.0'
    }

   1..4 | % {
   $VMNames = "HV";
   $LABConfig.VMs += @{
        VMName = "$VMNames$_";
        Configuration = 'S2D';
        ParentVHD = 'Win2019Core_G2.vhdx';
        SSDNumber = 2;
        SSDSize = 800GB;
        HDDNumber = 4;
        HDDSize = 4TB;
        MemoryStartupBytes = 20GB;
        NestedVirt = $True;
        StaticMemory = $True;
        VMProcessorCount = 6
       }
   }

   $LABConfig.VMs += @{
        VMName = "Management";
        Configuration = 'S2D';
        ParentVHD = 'Win2019_G2.vhdx';
        SSDNumber = 1;
        SSDSize = 50GB;
        MemoryStartupBytes = 4GB;
   ```

```
     NestedVirt = $false;
     StaticMemory = $false;
     VMProcessorCount = 4
}
```

5. Copy the **Scenario.ps1** and **MultiNodeConfig.psd1** files from **F:\WSLab-master\Scenarios\SDNExpress with Windows Admin Center** to **F:\WSLab-master\Scripts**.
6. From the Windows PowerShell ISE window, run the **F:\WSLab-master\Scripts\3_Deploy.ps1** script to provision **SDNExpress2019-DC** based on the **DC** VM and the remaining VMs for the SDN environment.

   > **Note**: Select **None** at the Telemetry prompt. The script should complete in about 7 minutes.

7. In the Windows PowerShell ISE window, open the **F:\WSLab-master\Scripts\Scenario.ps1** script, remove all content following the line **128**, starting from `# ENDING Run from Hyper-V Host ENDING #`, and then save the modified file as **Scenario_Part1.ps1**.

   > **Note**: You must run this part of the scenario script from the Hyper-V host.

8. In the Windows PowerShell ISE window, run the **F:\WSLab-master\Scripts\Scenario_Part1.ps1** script to configure the VMs that will host the lab environment. When prompted for the location of the parent VHDX for the SDN VMs, point to **F:\WSLab-master\Scripts\ParentDisks\Win2019Core_G2.vhdx**. When prompted for the **MultiNodeConfig.psd1** file, point to the file you copied to **F:\WSLab-master\Scripts**. When prompted for Windows Admin Center MSI, point to the downloaded Windows Installer file in the **F:\Source** folder.

   > **Note**: If the script fails with the message **Copy-VMFile : Failed to initiate copying files to the guest**, rerun the script.
   >
   > **Note**: The script should complete in about 15 minutes.
   >
   > **Note**: Ignore the error following the line **ScriptHalted** and message prompting to restart **SDNExpress2019-Management**. That's expected.

9. After the script completes, in the Windows PowerShell ISE window, run the following script to expand the size of the disks hosting drive **C** of the newly provisioned VMs that will host the SDN environment:

   ```
   $servers = @('SDNExpress2019-HV1','SDNExpress2019-HV2','SDNExpress2019-HV3','SDNExpress2019-HV4')
   $paths = (Get-VM -Name $servers | Get-VMHardDiskDrive | Where-Object {$_.ControllerLocation -eq 0}
   foreach ($path in $paths) { Resize-VHD -Path $path -SizeBytes 100GB }
   ```

10.5.3 Task 2: Deploy the SDN infrastructure VMs

Note: Make sure all of the VMs you provisioned in the previous task are running and that their operating system has been activated before you proceed to the next task. If that is not the case, start all of the VMs, sign in to each of them using the **CORP\LabAdmin** username and **LS1setup!** password and, from the elevated Command Prompt, run `slmgr -rearm`. The **DC** VM will require you to run `slmgr -rearm` and be restarted.

1. On the lab VM, use the Hyper-V Manager console to connect to the **SDNExpress2019-Management** VM. When prompted to sign in, provide the **CORP\LabAdmin** username and **LS1setup!** password.
2. Within the console session to the **SDNExpress2019-Management** VM, start Windows PowerShell ISE as Administrator and run the following script to expand the size of drive **C** of the VMs that will host the SDN environment:

   ```
   $servers = @('HV1','HV2','HV3','HV4')
   Invoke-Command -ComputerName $servers -ScriptBlock {
     $size = Get-PartitionSupportedSize -DriveLetter C
     Resize-Partition -DriveLetter C -Size $size.SizeMax
   }
   ```

3. On the lab VM, from the **script** pane of the Administrator: Windows PowerShell ISE window, run the following commands to download the following file:

   ```
   New-Item F:\Allfiles -itemtype directory -Force
   Invoke-Webrequest -Uri "https://raw.githubusercontent.com/MicrosoftLearning/WS-013T00-Azure-Stack-
   ```

4. Within the console session to the **SDNExpress2019-Management** VM, start File Explorer and navigate to the **C:\Library** folder.
5. Switch back to the lab VM and use the copy and paste functionality of the **Hyper-V** console session to copy **F:\WSLab-master\Scripts\Scenario.ps1** and **F:\Allfiles\SDNExpressModule.psm1** on the lab VM to **C:\Library** on the **SDNExpress2019-Management** VM.
6. Within the console session to the **SDNExpress2019-Management** VM, in the Administrator: Windows PowerShell ISE window, open the **C:\Library\Scenario.ps1** script, and comment out line 375 so it looks like so: `# Expand-Archive -Path C:\SDN-Master.zip -DestinationPath C:\Library`
7. Within the console session to the **SDNExpress2019-Management** VM, in the Administrator: Windows PowerShell ISE window, remove all content before the line 136, up to the line prior to `# Run from DC / VMM #`, and then save the modified file as **Scenario__Part2.ps1**.
8. Within the console session to the **SDNExpress2019-Management** VM, in the Administrator: Windows PowerShell ISE window, run the following command:

   ```
   Expand-Archive -Path C:\SDN-Master.zip -DestinationPath C:\Library
   Copy-Item -Path C:\Library\SDNExpressModule.psm1 -Destination C:\Library\SDN-master\SDNExpress\scr
   ```

   > **Note**: This part of the scenario script needs to be run from the management VM.

9. Within the console session to the **SDNExpress2019-Management** VM, in the Administrator: Windows PowerShell ISE window, run the newly saved **C:\Library\Scenario__Part2.ps1** script to configure the SDN VMs.

   > **Note**: Wait until the script completes before you proceed. The script should complete in about 90 minutes. Disregard any cluster validation errors.

10.5.4 Results

10.6 After completing this lab, you will have successfully provisioned the SDN infrastructure. --- lab: title: 'Lab B: Managing virtual networks by using Windows Admin Center and PowerShell' module: 'Module 4: Planning for and Implementing Azure Stack HCI Networking'

11 Lab B: Managing virtual networks by using Windows Admin Center and PowerShell

11.1 Scenario

Now you're ready to start testing the functionality of your Software-Defined Networking (SDN) environment. You'll start by provisioning virtual networks, deploying a few virtual machines (VMs) into them, and validating their connectivity within the same virtual network and between virtual networks.

11.2 Objectives

After completing this lab, you'll be able to manage virtual networks by using Windows Admin Center and PowerShell.

11.3 Estimated time: 90 minutes

11.4 Lab setup

To connect to the lab VM, follow the steps the lab hosting provider provides you.

11.5 Exercise 1: Managing virtual networks by using Windows Admin Center and PowerShell

11.5.1 Scenario

In this exercise, to provide connectivity between virtual networks, you will implement virtual network peering and use VMs to validate the peering configuration.

The main tasks for this exercise are as follows:

1. Connect to the SDN infrastructure by using Windows Admin Center.
2. Create virtual networks by using Windows Admin Center.
3. Create a storage volume on the hyperconverged cluster by using Windows Admin Center.
4. Create VMs by using Windows Admin Center.
5. Configure VMs.
6. Test network connectivity of VMs.
7. Connect virtual networks.
8. Test connectivity between peered virtual networks.

11.5.2 Task 1: Connect to the SDN infrastructure by using Windows Admin Center

1. From the lab VM, use the copy and paste functionality of the Hyper-V console session to copy F:\Source\ChromeStandaloneSetup64.exe to the C:\Library directory in the SDNExpress2019-Management VM. Within the console session to the SDNExpress2019-Management virtual machine (VM), switch to the File Explorer window displaying the content of the C:\Library folder and use the **ChromeStandaloneSetup64.exe** to install the Chrome browser. Also, install the **WindowsAdminCenter.exe** from C:\Library using all default settings except use port **9999**.

2. In the Chrome browser, navigate to the Windows Admin Center at `https://management:9999` and, if prompted to authenticate, sign in as **CORP\LabAdmin** with **LS1setup!** as the password.1. In the Chrome browser, navigate to the Windows Admin Center at `https://management:9999` and, if prompted to authenticate, sign in as **CORP\LabAdmin** with **LS1setup!** as the password.

   > **Note**: This URL designates the local installation of **Windows Admin Center** on the management VM. If Chrome initially refuses the connection to `https://management:9999` then try to connect to the URL with IE and then try Chrome again.

3. In the **Windows Admin Center** interface, add a connection to the `sddc01.corp.contoso.com` cluster and the Network Controller REST URI at `https://NCCLUSTER.corp.contoso.com`. If prompted, authenticate by using the **CORP\LabAdmin** and **LS1setup!** credentials.

11.5.3 Task 2: Create virtual networks by using Windows Admin Center

1. In the console session to the **SDNExpress2019-Management** VM, in the **Windows Admin Center** interface, open a connection to `sddc01.corp.contoso.com`.

2. On the `sddc01.corp.contoso.com` page, in the list of **Tools**, in the **Networking** section, select **Virtual switches**, and then review virtual switches on the members of the SDN cluster `sddc01.corp.contoso.com`.

3. Review settings of the first **sdnSwitch** on `hv1.corp.contoso.com` and note that you have the option of changing the **Load balancing algorithm** from **Hyper-V port** to **Dynamic**. Do not make any changes.

4. From the `sddc01.corp.contoso.com` page, navigate to the inventory of virtual networks.

5. From the **Inventory** tab, create the following virtual networks and subnets:

   Table 1: vnet-000 settings

Setting	Value
Name	vnet-000
Address Prefix	192.168.0.0/20

   Table 2: vnet-000 subnet-0 settings

Setting	Value
Name	subnet-0
Address Prefix	192.168.0.0/24

   Table 3: vnet-000 subnet-1 settings

Setting	Value
Name	subnet-1
Address Prefix	192.168.1.0/24

Table 4: vnet-100 settings

Setting	Value
Name	vnet-100
Address Prefix	192.168.96.0/20

Table 5: vnet-100 subnet-0 settings

Setting	Value
Name	subnet-0
Address Prefix	192.168.100.0/24

11.5.4 Task 3: Create a storage volume on the hyperconverged cluster by using Windows Admin Center

1. Copy the **ISO** image from the **F:\Source** folder on the lab VM to the **C:\Library** folder on the **SDNExpress2019-Management** VM.
2. In the console session on the **SDNExpress2019-Management** VM, in the browser window displaying the Windows Admin Center interface, from the `sddc01.corp.contoso.com` page, navigate to the inventory of storage volumes.
3. From the inventory panel of storage volumes of the `sddc01.corp.contoso.com` cluster, create the following volume:

   Table 6: VMStorage volume settings

Setting	Value
Name	VMStorage
Resiliency	Mirror-accelerated parity
Parity percentage	90% parity, 10% mirror
Size on hard disk drive (HDD)	512
Size units	GB

4. Upload the **ISO** file you copied to the **C:\Library** folder into the **VMStorage** volume.

   > **Note**: Wait for the upload to complete. If the ISO file doesn't upload correctly, then from **SDNExpress2019-Management** virtual machine (VM), connect to **\\HV3\c$** and paste the ISO file into the **\\HV3\c$\ClusterStorage\VMStorage** folder.

11.5.5 Task 4: Create VMs by using Windows Admin Center

1. In the console session on the **SDNExpress2019-Management** VM, in the Windows Admin Center interface, navigate to the inventory of VMs on the `sddc01.corp.contoso.com` cluster.
2. From the inventory panel of storage volumes of the `sddc01.corp.contoso.com` cluster, retain all other settings with their default values, and create VMs with the following settings:

   Table 7: vm-000 settings

Setting	Value
Name	vm-000
Generation	Generation 2 (Recommended)
Host	`hv3.corp.contoso.com`

Setting	Value
Path	C:\ClusterStorage\VMStorage
Virtual processor count	2
Enable nested virtualization	Disabled
Startup memory (GB)	2
Network adapter	sdnSwitch
Connect to virtual network	Enabled
Virtual network	vnet-000
Virtual subnet	subnet-0 [192.168.0.0/24]
IP Address	192.168.0.100
Storage	Create an empty virtual hard disk
Size (GB)	64
Operating system	Install an operating system from an image file (.iso)
Path	Path to the ISO file you copied to the C:\ClusterStorage\VMStorage volume in the previous

Table 8: vm-001 settings

Setting	Value
Name	vm-001
Generation	Generation 2 (Recommended)
Host	`hv3.corp.contoso.com`
Path	C:\ClusterStorage\VMStorage
Virtual processor count	2
Enable nested virtualization	Disabled
Startup memory (GB)	2
Network adapter	sdnSwitch
Connect to virtual network	Enabled
Virtual network	vnet-000
Virtual subnet	subnet-1 [192.168.1.0/24]
IP Address	192.168.1.100
Storage	Create an empty virtual hard disk
Size (GB)	64
Operating system	Install an operating system from an image file (.iso)
Path	Path to the ISO file you copied to the C:\ClusterStorage\VMStorage volume in the previous

Table 9: vm-100 settings

Setting	Value
Name	vm-100
Generation	Generation 2 (Recommended)
Host	`hv3.corp.contoso.com`
Path	C:\ClusterStorage\VMStorage
Virtual processor count	2
Enable nested virtualization	Disabled
Startup memory (GB)	2
Network adapter	sdnSwitch
Connect to virtual network	Enabled
Virtual network	vnet-100
Virtual subnet	subnet-0 [192.168.100.0/24]
IP Address	192.168.100.100
Storage	Create an empty virtual hard disk
Size (GB)	64
Operating system	Install an operating system from an image file (.iso)
Path	Path to the ISO file you copied to the C:\ClusterStorage\VMStorage volume in the previous

11.5.6 Task 5: Configure VMs

1. In the console session on the **SDNExpress2019-Management** VM, in the Windows Admin Center interface, on the inventory panel of VMs on the `sddc01.corp.contoso.com` cluster, identify the Microsoft Hyper-V host to which you deployed the VMs in the previous task (**HV3**).
2. In the console session on the **SDNExpress2019-Management** VM, start **Hyper-V Manager**, and use it to add the Hyper-V host you identified in the previous step to the console.
3. From the **Hyper-V Manager** console, establish console connections to the three VMs you deployed in the previous task.
4. Use the console connection to start the installation of **Windows Server 2019 Datacenter Evaluation** on each VM.

   Note: Wait for the operating system installation to complete on all three VMs.

5. Following the operating system installation, set the password of the built-in Administrator account to **Pa55w.rd** on each VM.
6. In the Windows Admin Center interface, on the inventory panel of VMs on the `sddc01.corp.contoso.com` cluster, shut down all three VMs.
7. In the Windows Admin Center interface, from the inventory panel of VMs on the `sddc01.corp.contoso.com` cluster, navigate to the network settings of each VM and configure their network adapters with the following settings:

   Table 10: vm-000 network adapter settings

Setting	Value
Connect to	Virtual Network
Virtual network	vnet-000
Virtual subnet	subnet-0 [192.168.0.0/24]
IP Address	192.168.0.100
MAC address type (Advanced)	Static

   Table 11: vm-001 network adapter settings

Setting	Value
Connect to	Virtual Network
Virtual network	vnet-000
Virtual subnet	subnet-1 [192.168.1.0/24]
IP Address	192.168.1.100
MAC address type	Static

   Table 12: vm-100 network adapter settings

Setting	Value
Connect to	Virtual Network
Virtual network	vnet-100
Virtual subnet	subnet-0 [192.168.100.0/24]
IP Address	192.168.100.100
MAC address type	Static

   Note: Network Controller automatically assigns the next available MAC address from its pool.

11.5.7 Task 6: Test network connectivity of VMs

1. On the **SDNExpress2019-Management** VM, in the Windows Admin Center, from the inventory panel of the `sddc01.corp.contoso.com` cluster, start the three VMs.
2. Use the Hyper-V Virtual Machine Connections to sign in to the three VMs you previously deployed in this

exercise, and from the Command Prompt, disable Windows Defender Firewall by running the following command:

```
powershell Set-NetFirewallProfile -Profile Domain,Public,Private -Enabled False
```

3. Switch back to Virtual Machine Connection to **vm-000**, and from the Command Prompt, run the following to test connectivity over WinRM to **vm-001**:

```
powershell Test-NetConnection -ComputerName 192.168.1.100 -Port 5985 -InformationLevel Detailed
```

4. Review the output and verify that the connection was successful.

   Note: This output is expected because Remote Management is enabled by default and **vm-000** and **vm-001** are on the same virtual network. The fact that they are not two different subnets does not have any significance in this case.

5. From the Command Prompt, run the following command to test connectivity over WinRM to **vm-100**:

```
powershell Test-NetConnection -ComputerName 192.168.100.100 -Port 5985 -InformationLevel Detailed
```

6. Review the output and verify that the connection failed.

   Note: This output is expected because although Remote Management is enabled by default, **vm-000** and **vm-100** are on different virtual networks. These virtual networks are not connected to each other at this point.

11.5.8 Task 7: Connect virtual networks

1. On the **SDNExpress2019-Management** VM, in the Windows Admin Center, navigate to the virtual network inventory panel of the `sddc01.corp.contoso.com` cluster. From that panel, display the settings of the **vnet-000** virtual network.
2. From the **vnet-000 Settings** panel, create a new peering with the following settings:

Table 13: vnet-000-to-vnet-100 peering settings

Setting	Value
Name	vnet-000-to-vnet-100
Virtual networks	vnet-100
Allow Virtual network access from 'vnet-000' to remote virtual network	Enabled
Allow forwarded traffic from 'vnet-000' to remote virtual network	Enabled
Allow Gateway Transit	Disabled

3. On the **vnet-000 Settings Peerings** panel, verify that the new peering is listed as **Connected** or **Initiated** in the **Peering Status** column.
4. In the Windows Admin Center, navigate to the virtual network inventory panel of the `sddc01.corp.contoso.com` cluster. From that panel, display the settings of the **vnet-100** virtual network.
5. From the **vnet-100 Settings** panel, create a new peering with the following settings:

Table 14: vnet-100-to-vnet-000 peering settings

Setting	Value
Name	vnet-100-to-vnet-000
Virtual networks	vnet-000
Allow Virtual network access from 'vnet-100' to remote virtual network	Enabled
Allow forwarded traffic from 'vnet-100' to remote virtual network	Enabled
Allow Gateway Transit	Disabled

6. Switch back to the **vnet-100 Settings Peerings** panel, and then verify that the new peering is listed as **Connected** in the **Peering Status** column.

11.5.9 Task 8: Test connectivity between peered virtual networks

> **Note**: For the change to take effect, the Network Controller Host agent on the Hyper-V host where the VMs reside must process the corresponding policy. To expedite the change, you will restart the agent and each of the VMs.

1. Within the Remote Desktop session to the **SDNExpress2019-Management** VM, switch to the browser window displaying the Windows Admin Center and on the upper left hand side of the page select **Windows Admin Center**.
2. Select the Hyper-V host (**HV3**) to which you deployed all three VMs.
3. On the page displaying the properties of the Hyper-V host, in the **Tools** list, select **Services**.
4. From the **Services** panel, restart the **NcHostAgent** service.
5. Within the console session to the **SDNExpress2019-Management** VM, switch to the **Hyper-V Manager** console displaying the Hyper-V host (**HV3**) to which you deployed all three VMs.
6. From the Hyper-V Manager console, restart the **vm-000**, **vm-001**, and **vm-100** VMs.
7. Switch to the **Virtual Machine Connection** to **vm-000** and sign in. From the Command Prompt, run the following to test connectivity over WinRM to **vm-100**:

   ```
   powershell Test-NetConnection -ComputerName 192.168.100.100 -Port 5985 -InformationLevel Detailed
   ```

8. Review the output and verify that the connection was successful.

   > **Note**: This output is expected because Remote Management is by default enabled and, at this point, while **vm-000** and **vm-100** are on different virtual networks, there are peering connections between them.

   > **Note**: If the connection fails, wait a few minutes, and try again.

11.5.10 Results

11.6 After completing this lab, you will have successfully managed virtual networks by using Windows Admin Center and PowerShell. --- lab: title: 'Lab C: Implementing SDN Access Control List by using Windows Admin Center' module: 'Module 4: Planning for and Implementing Azure Stack HCI Networking'

12 Lab C: Implementing SDN Access Control List by using Windows Admin Center

12.1 Scenario

As part of the security requirements within the Software-Defined Networking (SDN) environment, you need to be able to filter specific types of traffic between virtual network subnets. You intend to use the SDN functionality for this purpose, rather than relying exclusively on the operating system to perform this task.

12.2 Objectives

After completing this lab, you'll be able to implement SDN Access Control List by using Windows Admin Center.

12.3 Estimated time: 30 minutes

12.4 Lab setup

To connect to the lab VM, follow the steps the lab hosting provider provides you.

12.5 Exercise 1: Implementing SDN Access Control List by using Windows Admin Center

12.5.1 Scenario

In this exercise, you will create access control lists (ACLs) to filter specific types of traffic between virtual network subnets. You will use Windows Admin Center to create ACLs and to verify their functionality.

The main tasks for this exercise are as follows:

1. Create an ACL.
2. Assign the ACL to a subnet.
3. Verify functionality of the ACL.

12.5.2 Task 1: Create an ACL

1. Within the **SDNExpress2019-Management** VM, in the Windows Admin Center, on the `sddc01.corp.contoso.com` page, in the list of **Tools**, in the **Networking** section, select **Access control lists**.

2. On the **Access control lists** panel, from the **Inventory** tab, create a new ACL named **acl-100** with the following rules:

   The allow-all access rule settings are:

   - Name: **allow-all**
   - Priority: **1000**
   - Types: **Inbound**
   - Protocol: **All**
   - Source Address Prefix: *****
   - Source Port Range: *****
   - Destination Address Prefix: *****
   - Destination Port Range: *****
   - Action: **Allow**
   - Logging: **Enabled**

   The deny-winrm-from-vnet-000-subnet-0 access rule settings are:

   - Name: **deny-winrm-from-vnet-000-subnet-0**
   - Priority: **500**
   - Types: **Inbound**
   - Protocol: **TCP**
   - Source Address Prefix: **192.168.0.0/24**
   - Source Port Range: *****
   - Destination Address Prefix: *****
   - Destination Port Range: **5985,5986**
   - Action: **Deny**
   - Logging: **Enabled**

12.5.3 Task 2: Assign the ACL to a subnet

1. Within the console session to the **SDNExpress2019-Management** VM, start Windows PowerShell Integrated Scripting Environment (ISE) as Administrator and run the following script to list the properties of the virtual networks you created earlier in this exercise:

```
Import-Module NetworkController
$uri = 'https://NCCLUSTER.corp.contoso.com'
Get-NetworkControllerVirtualNetwork -ConnectionUri $uri
```

2. From the Windows PowerShell ISE window, run the following script to assign the ACL you created in the previous task to the first subnet (**subnet-0**) of the virtual network **vnet-100**:

```
$vnet2 = Get-NetworkControllerVirtualNetwork -ConnectionUri $uri -ResourceId 'vnet-100'
$acl = Get-NetworkControllerAccessControlList -ConnectionUri $uri -resourceid 'acl-100'
$vnet2.properties.subnets[0].Properties.AccessControlList = $acl
$subnet = Get-NetworkControllerVirtualSubnet -VirtualNetworkId $vnet2.ResourceId -ConnectionUri $u
New-NetworkControllerVirtualSubnet -ConnectionUri $uri -Properties $vnet2.Properties.Subnets[0].Pr
```

Note: Verify that the ACL assignment was created successfully.

3. Switch to the Windows Admin Center interface and refresh the browser page displaying the **Access control lists > acl-100** panel.
4. On the **Access Control List > acl-000** panel, in the **Related Tab** section, on the **Applied Virtual Subnets** tab, note the **subnet-0** entry of the **vnet-100** virtual network.

12.5.4 Task 3: Verify functionality of the ACL

Note: For the change to take effect, the Network Controller Host agent on the Microsoft Hyper-V host where the VMs reside must process the corresponding policy. To expedite the change, you will restart the agent and the third VM **vm-100**.

1. Within the console session to the **SDNExpress2019-Management** VM, switch to the browser window displaying the Windows Admin Center, and from the upper left hand side of the page, select **Windows Admin Center**.
2. Select the Hyper-V host (**HV3**) to which you deployed all three virtual machines.
3. On the page displaying the properties of the Hyper-V host, in the **Tools** list, select **Services**.
4. From the **Services** panel, restart the **NcHostAgent** service.
5. Within the console session to the **SDNExpress2019-Management** VM, switch to the **Hyper-V Manager** console displaying the Hyper-V host (**HV3**) to which you deployed all three VMs.
6. From the Hyper-V Manager console, restart the **vm-100** VM.
7. Switch to the **Virtual Machine Connection** to **vm-000**, from the Command Prompt, run the following command to test connectivity over WinRM to **vm-100**.

```
powershell Test-NetConnection -ComputerName 192.168.100.100 -Port 5985 -InformationLevel Detailed
```

8. Review the output and verify that the connection failed.

   Note: This output is expected because the Windows Remote Management traffic from **subnet-0** of **vnet-000** to which **vm-000** is attached is blocked by the newly created ACL assigned to **subnet-0** of **vnet-100**, to which **vm-100** is attached.

9. In the **Virtual Machine Connection** window to **vm-000**, from the Command Prompt, run the following command to test connectivity over Internet Control Message Protocol (ICMP) to **vm-100**:

```
ping 192.168.100.100
```

10. Review the output and verify that the connection was successful.

    Note: This output is expected because all other types of traffic (except for Remote Management) from **vnet-000** (including **subnet-0** to which **vm-000** is attached) are allowed by the newly created ACL assigned to **subnet-0** of **vnet-100**, to which the **vm-100** is attached.

11. Switch to the **Virtual Machine Connection** to **vm-001** and sign in. From the Command Prompt, run the following to test connectivity over WinRM to **vm-100**:

```
powershell Test-NetConnection -ComputerName 192.168.100.100 -Port 5985 -InformationLevel Detailed
```

12. Review the output and verify that the connection was successful.

    Note: This output is expected because Windows Remote Management traffic to **subnet-0** of **vnet-100**, to which the **vm-100** is attached is blocked only from **subnet-0** of **vnet-000**, and not from **subnet-1** to which **vm-001** is attached.

13. Switch to the **Virtual Machine Connection** to **vm-001** and sign in. From the Command Prompt, run the following command to test connectivity over WinRM to **vm-000**:

```
powershell Test-NetConnection -ComputerName 192.168.0.100 -Port 5985 -InformationLevel Detailed
```

14. Review the output and verify that the connection was successful.

    Note: This output is expected because the access control rule blocking Windows Remote Management traffic applies only to inbound traffic from **subnet-0** of **vnet-000**, not to outbound traffic from **subnet-0** of **vnet-100**, to which the **vm-100** is attached.

12.5.5 Results

12.6 After completing this lab, you will have successfully implemented SDN Access Control List by using Windows Admin Center. --- lab: title: 'Lab D: Implementing SDN Software Load Balancing with private virtual IP by using PowerShell' module: 'Module 4: Planning for and Implementing Azure Stack HCI Networking'

13 Lab D: Implementing SDN Software Load Balancing with private virtual IP by using PowerShell

13.1 Scenario

You need to configure virtual machines (VMs) on virtual networks that will serve load-balanced workloads accessible from within the datacenter hosting your Software-Defined Networking (SDN) infrastructure. In addition, you need to ensure that you will be able to configure VMs on virtual networks to connect to the internet and to accept inbound connectivity from your datacenter servers. Rather than relying on third-party load balancers, you intend to use SDN software load balancer for this purpose.

13.2 Objectives

After completing this lab, you'll be able to implement SDN Software Load Balancing by using Windows Admin Center and Windows PowerShell.

13.3 Estimated time: 120 minutes

13.4 Lab setup

To connect to the lab VM, follow the steps the lab hosting provider provides you.

13.5 Exercise 1: Implementing SDN Software Load Balancing by using Windows Admin Center and Windows PowerShell

13.5.1 Scenario

To meet the requirements of load-balanced workloads, you'll implement SDN Software Load Balancing by using Windows Admin Center and Windows PowerShell.

The main tasks for this exercise are as follows:

1. Review SDN virtual IP logical network configuration.
2. Install the Web Server role on VMs in a virtual network.
3. Configure an SLB private virtual IP address.
4. Verify the configuration of the SDN Software Load Balancing with private virtual IP address.
5. Configure outbound network address translation (NAT) by using SLB.
6. Verify the configuration of the SDN SLB outbound NAT.
7. Configure SLB-based traffic forwarding to a VM in a virtual network.
8. Verify connectivity to the VM in a virtual network via a public virtual IP address.

13.5.2 Task 1: Review SDN virtual IP logical network configuration

1. Within the **SDNExpress2019-Management** VM, in the Windows Admin Center, from the `sddc01.corp.contoso.com` page, display the logical network inventory.

2. In the logical network inventory, display the settings for the **PrivateVIP** logical network.

3. Review the logical network settings and note that it contains a single subnet with the IP address space of **10.20.0.0/24**.

4. Review the configuration of the subnet and note that it currently has **1** allocated IP address.

   > **Note**: You'll use an IP address from that range as a private virtual IP for connection to load balanced VMs on one of the virtual networks you created earlier in this lab.

5. In the logical network inventory, display the settings or the **PublicVIP** logical network.

6. Review the logical network settings and note that it contains a single subnet with the IP address space of **10.10.0.0/24**.
7. Review the configuration of the subnet and note that it currently has **1** allocated IP address.

   Note: You'll use an IP address from that range as a public virtual IP for outbound connectivity to the internet and for inbound connectivity from your datacenter.

13.5.3 Task 2: Install the Web Server role on VMs in a virtual network

1. Switch to the **Virtual Machine Connection** to **vm-000**. From the Command Prompt, run the following command to install the Web Server role.

   ```
   powershell Install-WindowsFeature -Name Web-Server
   ```

2. Use the procedure described in the previous step to install the Web server role on **vm-001**.

   Note: Wait for both installations to complete.

3. Switch to the **Virtual Machine Connection** to **vm-100**. From the Command Prompt, run the following to verify that the installation was successful.

   ```
   powershell Invoke-WebRequest -Uri 192.168.0.100 -UseBasicParsing
   powershell Invoke-WebRequest -Uri 192.168.1.100 -UseBasicParsing
   ```

   Note: Verify that in both cases you are receiving a response including **HTTP/1.1 200 OK**.

13.5.4 Task 3: Configure an SLB private virtual IP address

1. Switch to the console session to the **SDNExpress2019-Management** VM, from the Windows PowerShell Integrated Scripting Environment (ISE) window, run the following to create a load balancer object:

   ```
   Import-Module NetworkController
   $uri = 'https://NCCLUSTER.corp.contoso.com'
   $LBResourceId = 'lb-000'
   $LoadBalancerProperties = New-Object Microsoft.Windows.NetworkController.LoadBalancerProperties
   ```

2. From the same Windows PowerShell ISE window, run the following script to configure the private virtual IP:

   ```
   $vipIP = '10.20.0.100'
   $VIPLogicalNetwork = Get-NetworkControllerLogicalNetwork -ConnectionUri $uri -ResourceId 'PrivateVI
   $FrontEndIPConfig = New-Object Microsoft.Windows.NetworkController.LoadBalancerFrontendIpConfigura
   $FrontEndIPConfig.ResourceId = 'lb-000-fe-1'
   $FrontEndIPConfig.ResourceRef = "/loadBalancers/$LBResourceId/frontendIPConfigurations/$($FrontEndI

   $FrontEndIPConfig.Properties = New-Object Microsoft.Windows.NetworkController.LoadBalancerFrontendI
   $FrontEndIPConfig.Properties.Subnet = New-Object Microsoft.Windows.NetworkController.Subnet
   $FrontEndIPConfig.Properties.Subnet.ResourceRef = $VIPLogicalNetwork.Properties.Subnets[0].Resource
   $FrontEndIPConfig.Properties.PrivateIPAddress = $vipIP
   $FrontEndIPConfig.Properties.PrivateIPAllocationMethod = 'Static'
   $LoadBalancerProperties.FrontEndIPConfigurations += $FrontEndIPConfig
   ```

   Note: The virtual IP belongs to the IP address range of the subnet of the logical network you identified in the first task of this exercise.

3. From the same Windows PowerShell ISE window, run the following to configure the back-end address pool, which contains the Dynamic IPs assigned to the load-balanced set of VMs:

   ```
   $BackEndAddressPool = New-Object Microsoft.Windows.NetworkController.LoadBalancerBackendAddressPool
   $BackEndAddressPool.ResourceId = 'lb-000-be-1'
   $BackEndAddressPool.ResourceRef = "/loadBalancers/$LBResourceId/backendAddressPools/$($BackEndAddre
   $BackEndAddressPool.Properties = New-Object Microsoft.Windows.NetworkController.LoadBalancerBacken
   $LoadBalancerProperties.backendAddressPools += $BackEndAddressPool
   ```

4. From the same Windows PowerShell ISE window, run the following script to define a health probe that the load balancer will use to determine the health state of the back-end pool members:

```
$Probe = New-Object Microsoft.Windows.NetworkController.LoadBalancerProbe
$Probe.ResourceId = 'lb-000-hp-1'
$Probe.ResourceRef = "/loadBalancers/$LBResourceId/Probes/$($Probe.ResourceId)"
$Probe.properties = New-Object Microsoft.Windows.NetworkController.LoadBalancerProbeProperties
$Probe.properties.Protocol = 'HTTP'
$Probe.properties.Port = '80'
$Probe.properties.RequestPath = '/'
$Probe.properties.IntervalInSeconds = 5
$Probe.properties.NumberOfProbes = 5
$LoadBalancerProperties.Probes += $Probe
```

5. From the same Windows PowerShell ISE window, run the following script to define a load balancing rule to distribute traffic that arrives at the front-end IP to back-end IPs. In this case, the back-end pool receives Transmission Control Protocol (TCP) traffic to port **80**.

```
$Rule = New-Object Microsoft.Windows.NetworkController.LoadBalancingRule
$Rule.ResourceId = 'web-000'
$Rule.Properties = New-Object Microsoft.Windows.NetworkController.LoadBalancingRuleProperties
$Rule.Properties.FrontEndIPConfigurations += $FrontEndIPConfig
$Rule.Properties.backendaddresspool = $BackEndAddressPool
$Rule.Properties.protocol = 'TCP'
$Rule.Properties.FrontEndPort = 80
$Rule.Properties.BackEndPort = 80
$Rule.Properties.IdleTimeoutInMinutes = 4
$Rule.Properties.Probe = $Probe
$LoadBalancerProperties.loadbalancingRules += $Rule
```

6. From the same Windows PowerShell ISE window, run the following command to apply the change by adding the load balancer configuration to Network Controller:

```
$LoadBalancerResource = New-NetworkControllerLoadBalancer -ConnectionUri $URI -ResourceId $LBResou
```

7. From the same Windows PowerShell ISE window, run the following commands to retrieve the reference to the load balancer object:

```
$lbresourceid = 'lb-000'
$lb = Get-NetworkControllerLoadBalancer -ConnectionUri $uri -ResourceID $LBResourceId -PassInnerEx
```

8. From the same Windows PowerShell ISE window, run the following command to identify values of the **ResourceId** property of network interfaces assigned to the VMs you created earlier in the lab:

```
Get-NetworkControllerNetworkInterface -ConnectionUri $uri | Select-Object ResourceId
```

9. From the same Windows PowerShell ISE window, run the following script to add the network interface of **vm-000** to the back-end pool of the load balancer **lb-000**:

```
$nic1 = Get-NetworkControllerNetworkInterface -ConnectionUri $uri -ResourceId 'vm-000_Net_Adapter_
$nic1.properties.IpConfigurations[0].properties.LoadBalancerBackendAddressPools += $lb.properties.
New-NetworkControllerNetworkInterface -ConnectionUri $uri -ResourceId 'vm-000_Net_Adapter_0' -prop
```

10. From the same Windows PowerShell ISE window, run the following script to add the network interface of **vm-001** to the back-end pool of the load balancer **lb-000**:

```
$nic2 = Get-NetworkControllerNetworkInterface -ConnectionUri $uri -ResourceId vm-001_Net_Adapter_0
$nic2.properties.IpConfigurations[0].properties.LoadBalancerBackendAddressPools += $lb.properties.
New-NetworkControllerNetworkInterface -ConnectionUri $uri -ResourceId 'vm-001_Net_Adapter_0' -prop
```

13.5.5 Task 4: Verify the configuration of the SDN Software Load Balancing with private virtual IP

1. Within the **SDNExpress2019-Management** VM, switch to the browser window displaying the **Logical subnet > 10.20.0.0_24** panel within the Windows Admin Center interface.

2. Refresh the browser page, review the updated configuration, and note that it currently has **2** allocated IP addresses.

3. In the console session to the **SDNExpress2019-Management** VM, open a new browser window.

4. In the browser window, navigate to `http://10.20.0.100` and verify that you can access the default Microsoft Internet Information Services (IIS) home page.

5. Within the console session to the **SDNExpress2019-Management** VM, switch to the Windows PowerShell ISE window, and run the following command from the **console** pane to identify the Border Gateway Patrol (BGP) router information, which is hosted on the **SDNExpress2019-DC** VM:

   ```
   Invoke-Command -ComputerName DC -ScriptBlock {Get-BgpRouter}
   ```

   > **Note**: Note that the BGP peers include two Gateway VMs and the two multiplexer (MUX) VMs.

6. In the Windows PowerShell ISE window, and run the following from the **console** pane to identify the BGP route information, (with the router hosted on the **SDNExpress2019-DC** VM):

   ```
   Invoke-Command -ComputerName DC -ScriptBlock {Get-BgpRouteInformation}
   ```

   > **Note**: Note that the output includes two routes to the private virtual IP you configured in this exercise (one per MUX) and that each route was learned from the corresponding MUX VM.

13.5.6 Task 5: Configure outbound NAT by using SLB

1. In the console session to the **SDNExpress2019-Management** VM, from the Windows PowerShell ISE window, run the following to create a load balancer object:

   ```
   Import-Module NetworkController
   $uri = 'https://NCCLUSTER.corp.contoso.com'
   $LBResourceId = 'lb-nat-outbound-100'
   $LoadBalancerProperties = New-Object Microsoft.Windows.NetworkController.LoadBalancerProperties
   ```

2. From the same Windows PowerShell ISE window, run the following to create the load balancer, its front-end IP, and the back-end pool:

   ```
   $vipIP = '10.10.0.100'
   $vipLogicalNetwork = Get-NetworkControllerLogicalNetwork -ConnectionUri $uri -resourceid 'PublicVI

   $FrontEndIPConfig = new-object Microsoft.Windows.NetworkController.LoadBalancerFrontendIpConfigura
   $FrontEndIPConfig.ResourceId = 'fe-100'
   $FrontEndIPConfig.ResourceRef = "/loadBalancers/$LBResourceId/frontendIPConfigurations/$($FrontEnd
   $FrontEndIPConfig.Properties = new-object Microsoft.Windows.NetworkController.LoadBalancerFrontend
   $FrontEndIPConfig.Properties.Subnet = new-object Microsoft.Windows.NetworkController.Subnet
   $FrontEndIPConfig.Properties.Subnet.ResourceRef = $vipLogicalNetwork.Properties.Subnets[0].Resourc
   $FrontEndIPConfig.Properties.PrivateIPAddress = $vipIP
   $FrontEndIPConfig.Properties.PrivateIPAllocationMethod = 'Static'
   $LoadBalancerProperties.FrontEndIPConfigurations += $FrontEndIPConfig

   $BackEndAddressPool = new-object Microsoft.Windows.NetworkController.LoadBalancerBackendAddressPoo
   $BackEndAddressPool.ResourceId = 'bepool-100'
   $BackEndAddressPool.ResourceRef = "/loadBalancers/$LBResourceId/backendAddressPools/$($BackEndAddr
   $BackEndAddressPool.Properties = new-object Microsoft.Windows.NetworkController.LoadBalancerBacken

   $LoadBalancerProperties.backendAddressPools += $BackEndAddressPool
   ```

   > **Note**: The virtual IP belongs to the IP address range of the subnet of the logical network you identified in the first task of this exercise.

3. From the same Windows PowerShell ISE window, run the following to define the outbound NAT rule:

   ```
   $OutboundNAT = new-object Microsoft.Windows.NetworkController.LoadBalancerOutboundNatRule
   $OutboundNAT.ResourceId = 'outbound-nat-100'

   $OutboundNAT.properties = new-object Microsoft.Windows.NetworkController.LoadBalancerOutboundNatRu
   $OutboundNAT.properties.frontendipconfigurations += $FrontEndIPConfig
   $OutboundNAT.properties.backendaddresspool = $BackEndAddressPool
   $OutboundNAT.properties.protocol = 'ALL'
   ```

4. From the same Windows PowerShell ISE window, run the following command to apply the change by adding the load balancer configuration to Network Controller:

```
$LoadBalancerResource = New-NetworkControllerLoadBalancer -ConnectionUri $uri -ResourceId $LBResou
```

5. From the same Windows PowerShell ISE window, run the following command to retrieve the reference to the load balancer object:

```
$lbresourceid = 'lb-nat-outbound-100'
$lb = Get-NetworkControllerLoadBalancer -ConnectionUri $uri -ResourceID $LBResourceId -PassInnerEx
```

6. From the same Windows PowerShell ISE window, run the following command to identify values of the **ResourceId** properties of network interfaces assigned to the VMs you previously created in the lab:

```
Get-NetworkControllerNetworkInterface -ConnectionUri $uri | Select-Object ResourceId
```

7. From the same Windows PowerShell ISE window, run the following script to add the network interface of **vm-100** to the back-end pool of the load balancer **lb-nat-outbound-100**:

```
$nic1 = Get-NetworkControllerNetworkInterface -ConnectionUri $uri -ResourceId 'vm-100_Net_Adapter_
$nic1.properties.IpConfigurations[0].properties.LoadBalancerBackendAddressPools += $lb.properties.
New-NetworkControllerNetworkInterface -ConnectionUri $uri -ResourceId 'vm-100_Net_Adapter_0' -prop
```

13.5.7 Task 6: Verify the configuration of the SDN Software Load Balancing outbound NAT

1. Within the console session to the **SDNExpress2019-Management** VM, switch to the browser window displaying within Windows Admin Center, and navigate to the **Logical subnet > 10.10.0.0__24** panel in the **PublicVIP** section.

2. Refresh the browser page, review the updated configuration, and note that it currently has **2** allocated IP addresses.

3. Switch to the lab VM and, from the console pane of the Administrator: Windows PowerShell ISE window, run the following to install the Web Server role:

```
Install-WindowsFeature -Name Web-Server
```

> **Note**: Wait for the installation to complete.

4. On the lab VM, from the console pane of the Administrator: Windows PowerShell ISE window, run the following to identify the local IP configuration:

```
Get-NetIPConfiguration
```

> **Note**: Review the output of the cmdlet you ran in the previous step and identify the IP address of its network interace which is **NOT** used for internal NAT. You will need it in this task.

5. On the lab VM, open a new browser window and navigate to the IP address you identified in the previous step and verify that you can access the default IIS home page.

> **Note**: This is the default web site installed on the lab VM, accessible via its private IP address. Verify that the Windows Firewall on the lab VM is allowing inbound traffic on port 80 for all network profiles.

6. Record the IP address and switch back to the console session to the **SDNExpress2019-Management** VM.

7. Within the console session to the **SDNExpress2019-Management** VM, switch to the **Virtual Machine Connection** window to **vm-100**, and from the Command Prompt, run the following command to test connectivity to the public IP address you identified in the previous step. Replace the `[ip_address]` placeholder with the IP address you recorded in the previous step:

```
powershell Invoke-WebRequest -Uri [ip_address] -UseBasicParsing
```

> **Note**: Verify that you receive a response with the **Status Code** of **200**.

8. In the Windows PowerShell ISE window, run the following from the **console** pane to identify the BGP route information (with the router hosted on the **SDNExpress2019-DC** VM):

```
Invoke-Command -ComputerName DC -ScriptBlock {Get-BgpRouteInformation}
```

> **Note**: Note that the output includes two routes to the public virtual IP you configured in this exercise (one per MUX) and that each route was learned from the corresponding MUX VM.

13.5.8 Task 7: Configure SLB-based traffic forwarding to a VM in a virtual network

1. Within the console session to the **SDNExpress2019-Management** VM, switch to the Windows PowerShell ISE window, and run the following commands to create a public IP address object referencing a public virtual IP:

```
$publicIPProperties = new-object Microsoft.Windows.NetworkController.PublicIpAddressProperties
$publicIPProperties.ipaddress = '10.10.0.200'
$publicIPProperties.PublicIPAllocationMethod = 'static'
$publicIPProperties.IdleTimeoutInMinutes = 4
$publicIP = New-NetworkControllerPublicIpAddress -ResourceId 'vm-100-pip' -Properties $publicIPProp
```

2. From the same Windows PowerShell ISE window, run the following commands to assign the newly created public IP address object to the network interface of the **vm-100** VM:

```
$nic = Get-NetworkControllerNetworkInterface  -connectionuri $uri -resourceid 'vm-100_Net_Adapter_
$nic.properties.IpConfigurations[0].Properties.PublicIPAddress = $publicIP
New-NetworkControllerNetworkInterface -ConnectionUri $uri -ResourceId $nic.ResourceId -Properties $
```

13.5.9 Task 8: Verify connectivity to the VM in a virtual network via a public virtual IP

1. In the Windows PowerShell ISE window, and run the following command from the **console** pane to identify the BGP route information, with the router hosted on the **SDNExpress2019-DC** VM:

```
Invoke-Command -ComputerName DC -ScriptBlock {Get-BgpRouteInformation}
```

> **Note**: Note that the output includes two routes to the public virtual IP you configured in this exercise (one per MUX) and that each route was learned from the corresponding MUX VM.

> **Note**: If the routes are not displayed yet, you might need to wait a few minutes.

2. Within the console session to the **SDNExpress2019-Management** VM, the Windows PowerShell ISE window, from the **console** pane, run the following to determine whether you have connectivity to the **vm-100** via the IP address allocated from the **PublicVIP** subnet.

```
Test-NetConnection -ComputerName 10.10.0.200 -Port 5985 -InformationLevel Detailed
```

3. Verify that the connection attempt was successful.

4. Within the console session to the **SDNExpress2019-Management** VM, in the Windows PowerShell ISE window, from the **console** pane, run the following to establish a PowerShell Remoting session to the **vm-100** VM via the IP address allocated from the **PublicVIP** subnet.

```
$username = 'Administrator'
$password = ConvertTo-SecureString -String 'Pa55w.rd' -AsPlainText -Force
$creds = New-Object -TypeName System.Management.Automation.PSCredential -ArgumentList $username,$pa
Enter-PSSession -ComputerName 10.10.0.200 -Credential $creds
```

5. After the session is established, from the **console** pane in the Windows PowerShell ISE window, from the `[10.10.0.200]: PS C:\Users\Administrator\Documents>` prompt, run `ipconfig` and verify that the output displays the IP configuration of the **vm-100** VM, with the IP address of **192.168.100.100**.

13.5.10 Task 9: Deprovision the lab resources

1. Within the Remote Desktop session to lab VM, start Windows PowerShell ISE as Administrator.

2. From the Windows PowerShell ISE window, run the **F:\WSLab-master\Scripts\Cleanup.ps1** script to remove all VMs provisioned in this lab.

13.5.11 Results

13.6 After completing this lab, you would have deployed SDN by using PowerShell, configured routing and management for the SDN lab environment, managed virtual networks by using Windows Admin Center and PowerShell, implemented SDN Access Control List by using Windows Admin Center, implemented SDN Software Load Balancing with private virtual IP by using PowerShell, and deprovisioned the lab environment.

13.7 lab: title: 'Lab: Using Windows Admin Center in hybrid scenarios' type: 'Answer Key' module: 'Module 2: Operating and maintaining Azure Stack HCI'

14 Lab answer key: Using Windows Admin Center in hybrid scenarios

14.1 Exercise 1: Provisioning the lab environment by using PowerShell

14.1.1 Task 1: Prepare the lab artifacts

1. From the lab virtual machine (VM), start Windows PowerShell ISE as Administrator.
2. In the Administrator: Windows PowerShell ISE window, from the console pane, run the following to remove the **Zone.Identifier** alternate data stream, which has a value of **3**, indicating that it was downloaded from the Internet:

```
Get-ChildItem -Path F:\WSLab-master\ -File -Recurse | Unblock-File
```

14.1.2 Task 2: Deploy the lab infrastructure

1. On the lab VM, in the console pane of the PowerShell ISE window, run the following to set the current directory:

```
Set-Location -Path F:\WSLab-master\Scripts
```

2. In the script pane of the PowerShell ISE window, run the following to rename **Scenario.ps1** and **LabConfig.ps1**:

```
Move-Item -Path '.\LabConfig.ps1' -Destination '.\LabConfig.m2l0.ps1' -Force -ErrorAction Silently
Move-Item -Path '.\Scenario.ps1' -Destination '.\Scenario.m2l0.ps1' -Force -ErrorAction SilentlyCon
```

3. In the script pane of the PowerShell ISE window, run the following to copy **Scenario.ps1** and **Labconfig.ps1** files from **F:\WSLab-master\Scenarios\S2D and Cloud Services Onboarding** to the current directory:

```
Copy-Item -Path 'F:\WSLab-master\Scenarios\S2D and Cloud Services Onboarding\Scenario.ps1' -Destina
Copy-Item -Path 'F:\WSLab-master\Scenarios\S2D and Cloud Services Onboarding\Labconfig.ps1' -Desti
```

4. In the script pane of the PowerShell ISE window, open the **F:\WSLab-master\Scripts\LabConfig.ps1** file, in the first line, replace `Prefix = 'WSLab-'` with `Prefix = 'WSLabOnboard-'`, save the changes, and then close the file.
5. In the PowerShell ISE window, open and run the **F:\WSLab-master\Scripts\3_Deploy.ps1** script to VMs for the lab environment.

> **Note**: For the Telemetry Level prompt, select the default setting of None. The script should complete in about seven minutes. For the prompt to start the VMs, select All to Start the VMs. When prompted with Press enter to continue, select **Enter**.

14.2 Exercise 2: Integrating hyperconverged infrastructure with Azure services

14.2.1 Task 1: Prepare the lab infrastructure VMs for integration with Azure services

1. On the lab VM, from the console pane of the PowerShell ISE window, run the following to start all of the lab infrastructure VMs:

```
Get-VM | Where-Object Name -ne 'WSLabOnboard-DC' | Start-VM
```

2. On the lab VM, start the Hyper-V Manager console and select the node representing the lab VM; in the list of VMs, right-click or access the context menu on the **WSLabOnboard-DC** entry; and then select **Connect**. When you receive a prompt, select **Connect**, and then sign in by using **CORP\LabAdmin** as the username and **LS1setup!** as the password.

3. Within the console session to the **WSLabOnboard-DC** VM, start File Explorer, and then create a folder **C:\Library**.

4. Switch back to the lab VM and use the copy and paste functionality of the Hyper-V console session to copy **F:\WSLab-master\Scripts\Scenario.ps1** on the lab VM to **C:\Library** on the **WSLabOnboard-DC** VM.

5. Within the console session to the **WSLabOnboard-DC** VM, start Windows PowerShell ISE as Administrator.

6. Within the console session to the **WSLabOnboard-DC** VM, in the PowerShell ISE window, open the **C:\Library\Scenario.ps1** file in the script pane, select the first part of the script between the lines **1** and **25** marked as **#region Prereqs**, and then run that part of the script either by selecting the **F8** key or selecting the **Run selection** button in the toolbar of the PowerShell ISE window.

   Note: This part of the script installs prerequisites that allow subsequent parts of the script to run, including Remote Server Administration Tools and Azure PowerShell modules.

   Note: Wait for the script to complete. Ignore any errors regarding **Login-AZaccount**.

7. Within the console session to the **WSLabOnboard-DC** VM, in the PowerShell ISE window, in the **C:\Library\Scenario.ps1** file in the script pane, select the second part of the script between the lines **27** and **61** marked as **#region Install Windows Admin Center in a GW mode**, and then run that part of the script either by selecting the **F8** key or selecting the **Run selection** button in the toolbar of the PowerShell ISE window.

   Note: This part of the script installs Windows Admin Center in the gateway mode on the **WACGW** VM.

   Note: Ignore error messages regarding aborted I/O operation.

8. Install the Microsoft Edge based on Chromium browser.

9. Within the console session to the **WSLabOnboard-DC** VM, complete the Microsoft Edge installation by selecting **Get started**, following subsequent prompts, and then closing the browser window.

14.2.2 Task 2: Provision a Storage Spaces Direct cluster within the lab environment

1. Switch to the lab VM and in the PowerShell ISE window, from the console pane, run the following to shut down the VMs that will serve as nodes of the Storage Spaces Direct cluster in this lab:

```
$VMs = @('WSLabOnboard-S2D1','WSLabOnboard-S2D2','WSLabOnboard-S2D3','WSLabOnboard-S2D4')
Stop-VM -VMName $VMs -Force
```

2. On the lab VM, in the PowerShell ISE window, from the console pane, run the following to configure nested virtualization for the VMs that will serve as nodes of the Storage Spaces Direct cluster in this lab:

```
Set-VMProcessor -VMName $VMs -ExposeVirtualizationExtensions $true
```

3. On the lab VM, in the PowerShell ISE window, from the console pane, run the following to configure static memory for the VMs that will serve as nodes of the Storage Spaces Direct cluster in this lab:

```
Set-VM $VMs -ProcessorCount 2 -StaticMemory -MemoryStartupBytes 4GB
```

   Note: This is not required for nested virtualization but mitigates problems with memory during startup.

4. On the lab VM, in the PowerShell ISE window, from the console pane, run the following to start the VMs that will serve as nodes of the Storage Spaces Direct cluster in this lab:

```
Start-VM -VMName $VMs
```

5. Switch to the console session to the **WSLabOnboard-DC** VM. In the PowerShell ISE window, open a new tab in the script pane, paste the following script, and then run it to install Windows Server 2019 roles and features necessary to provision Storage Spaces Direct cluster on the four VMs in the lab environment (**S2D1**, **S2D2**, **S2D3**, and **S2D4**):

```
$servers = @('S2D1','S2D2','S2D3','S2D4')
$features = 'Hyper-V', 'Failover-Clustering', 'Data-Center-Bridging', 'RSAT-Clustering-PowerShell'
Invoke-Command ($servers) {
  Install-WindowsFeature -Name $using:features
}
```

6. Within the console session to the **WSLabOnboard-DC** VM, in the PowerShell ISE window, open a new tab in the script pane, paste the following script, and then run it to complete installation of server roles and features by restarting the four VMs in the lab environment (**S2D1**, **S2D2**, **S2D3**, and **S2D4**):

```
Invoke-Command ($servers) {
  Restart-Computer -Force
}
```

   > **Note**: Wait a few minutes until the operating system in all four VMs is running.

7. Within the console session to the **WSLabOnboard-DC** VM, in the PowerShell ISE window, open a new tab in the script pane, paste the following script, and then run it to configure storage to prepare for provisioning of a Storage Spaces Direct cluster on the four VMs in the lab environment (**S2D1**, **S2D2**, **S2D3**, and **S2D4**):

```
Invoke-Command ($servers) {
  Update-StorageProviderCache
  Get-StoragePool | ? IsPrimordial -eq $false | Set-StoragePool -IsReadOnly:$false -ErrorAction Si
  Get-StoragePool | ? IsPrimordial -eq $false | Get-VirtualDisk | Remove-VirtualDisk -Confirm:$fal
  Get-StoragePool | ? IsPrimordial -eq $false | Remove-StoragePool -Confirm:$false -ErrorAction Si
  Get-PhysicalDisk | Reset-PhysicalDisk -ErrorAction SilentlyContinue
  Get-Disk | ? Number -ne $null | ? IsBoot -ne $true | ? IsSystem -ne $true | ? PartitionStyle -ne
     $_ | Set-Disk -isoffline:$false
     $_ | Set-Disk -isreadonly:$false
     $_ | Clear-Disk -RemoveData -RemoveOEM -Confirm:$false
     $_ | Set-Disk -isreadonly:$true
     $_ | Set-Disk -isoffline:$true
 }
 Get-Disk | Where Number -Ne $Null | Where IsBoot -Ne $True | Where IsSystem -Ne $True | Where Par
} | Sort -Property PsComputerName, Count
```

8. Within the console session to the **WSLabOnboard-DC** VM, in the PowerShell ISE window, from the console pane, run the following cmdlet to perform cluster validation for the four VMs in the lab environment (**S2D1**, **S2D2**, **S2D3**, and **S2D4**):

```
Test-Cluster -Node 'S2D1','S2D2','S2D3','S2D4' -Include 'Storage Spaces Direct', 'Inventory', 'Netw
```

   > **Note**: In order to run Test-Cluster from the **WSLabOnboard-DC** VM, you will need to install the Failover Clustering feature and restart the **WSLabOnboard-DC** VM. Ignore cluster validation errors. That's expected.

9. Within the console session to the **WSLabOnboard-DC** VM, in the PowerShell ISE window, from the console pane, run the following cmdlet to create a new cluster consisting of the four VMs in the lab environment (**S2D1**, **S2D2**, **S2D3**, and **S2D4**):

```
New-Cluster -Name 'S2DCL1' -Node 'S2D1','S2D2','S2D3','S2D4' -NoStorage
```

   > **Note**: Wait for the cluster to be provisioned.

14.2.3 Task 3: Configure Cloud Witness quorum for the Storage Spaces Direct cluster

1. Within the console session to the **WSLabOnboard-DC** VM, start the Microsoft Edge based on Chromium browser, navigate to the Azure portal and, when you receive a prompt, sign in with the Owner or Contributor role in the Azure subscription you will be using in this lab.
2. In the Azure portal, in the **Search resources, services, and docs** text box at the top of the Azure portal page, enter **Storage accounts**, and then select the **Enter** key.
3. On the **Storage accounts** blade, select **+ Add**.
4. On the **Basics** tab of the **Create storage account** blade, specify the following settings (leave others with their default values):

Table 1: Storage account settings

Setting	Value
Subscription	the name of the Azure subscription you are using in this lab
Resource group	WS013-02-RG
Storage account name	any globally unique name between 3 and 24 in length consisting of letters and digits
Location	the name of an Azure region in proximity to the location of the lab environment
Performance	Standard
Account kind	Storage (general purpose v1)
Replication	Locally redundant storage (LRS)

5. On the **Basics** tab of the **Create storage account** blade, select **Review + Create**, wait for the validation process to complete, and then select **Create**.

   Note: Wait for the Storage account to be created. This should take about two minutes.

6. In the Azure portal, in the **Search resources, services, and docs** text box at the top of the Azure portal page, enter **Resource groups**, and then select the **Enter** key.
7. On the **Resource groups** blade, in the list of resource group, select the **WS013-02-RG** entry.
8. On the **WS013-02-RG** resource group blade, in the list of resources, select the entry representing the newly created storage account.
9. On the storage account blade, in the **Settings** section, select **Access keys**.
10. From the blade displaying access keys, copy the values of **Storage account name** and **key1** into Notepad.

    Note: You will need both values later in this task.

11. Within the console session to the **WSLabOnboard-DC** VM, start another instance of the Microsoft Edge based on Chromium browser, and then navigate to `https://wacgw.corp.contoso.com`. When you receive a prompt, sign in by using **CORP\LabAdmin** as the user name and **LS1setup!** as the password.

    Note: Select **Continue** if you receive an error that the connection is not secure.

12. In the **Windows Admin Center** interface, on the **All connections** page, select **+ Add**. On the **Add resources** panel, in the **Server clusters** tile, select **Add**. In the **Cluster name** text box, enter `S2DCL1.corp.contoso.com`and select **Use another account for this connection**. In the **Username** text box, enter **CORP\LabAdmin**, in the **Password** text box, enter **LS1setup!**; select **Connect with account**, and then select **Add**.
13. Back on the **All connections** page, select the `S2DCL1.corp.contoso.com` entry.
14. On the `S2DCL1.corp.contoso.com` page, examine the **Overview** panel, and then select **Settings**.
15. On the **Settings** panel, in the **Cluster** section, select **Witness**, and then in the **Witness type** drop down list, select **Cloud witness**.
16. In the **Azure storage account name** text box, paste the value of the **Storage account name** you copied earlier in this task.
17. In the **Azure storage account key** text box, paste the value of the **key1** you copied earlier in this task.
18. Select **Save** and when you receive a prompt to enable CredSSP, select **Yes**.

14.2.4 Task 4: Enable Storage Spaces Direct on the cluster

1. Within the console session to the **WSLabOnboard-DC** VM, switch to the PowerShell ISE window and from the console pane, run the following cmdlet to enable Storage Spaces Direct on the newly created cluster (when you receive a prompt whether to proceed, select **Yes to All**).

   ```
   Enable-ClusterStorageSpacesDirect -CimSession 'S2DCL1'
   ```

   Note: Disregard an error message regarding **No disks found to be used for cache**.

2. Switch back to the browser window displaying the **Windows Admin Center** interface; on the **Settings** panel of `S2DCL1.corp.contoso.com` page, select **Storage Spaces and Pools**; and then examine its settings.

Note: You might need to refresh the browser page to connect to the cluster.

14.2.5 Task 5: Provision Azure Log Analytics workspace and Azure Log Analytics gateway

1. Within the console session to the **WSLabOnboard-DC** VM, switch to the browser window displaying the Azure portal, and then select the **Cloud Shell** icon in the toolbar of the portal interface.
2. If you receive a prompt to select either **Bash** or **PowerShell**, select **PowerShell**.

   Note: If this is the first time you are starting **Cloud Shell** and you are presented with the **You have no storage mounted** message, select the subscription you are using in this lab, and then select **Create storage**.

3. From the Cloud Shell pane, run the following to register the **Microsoft.Insights** and **Microsoft.AlertsManagement** resource providers:

```
Register-AzResourceProvider -ProviderNamespace Microsoft.Insights
Register-AzResourceProvider -ProviderNamespace Microsoft.AlertsManagement
```

4. From the Cloud Shell pane, run the following to verify that the registration was successful:

```
Get-AzResourceProvider -ProviderNamespace Microsoft.Insights
Get-AzResourceProvider -ProviderNamespace Microsoft.AlertsManagement
```

   Note: Wait until the **RegistrationState** is listed as **Registered**.

5. Within the console session to the **WSLabOnboard-DC** VM, switch to the PowerShell ISE window. In the **C:\Library\Scenario.ps1** file in the script pane, in line **77**, replace **OutpuMode** with **OutputMode**, and then save the change.

   Note: Run **Install-Module AZ** on the **WSLabOnboard-DC** VM prior to performing the next step.

6. Within the console session to the **WSLabOnboard-DC** VM, in the PowerShell ISE window, in the **C:\Library\Scenario.ps1** file in the script pane, select the fourth part of the script between the lines **74** and **105** marked as **#region Connect to Azure and create Log Analytics workspace if needed**, and then run that part of the script either by selecting the **F8** key or selecting the **Run selection** button in the toolbar of the PowerShell ISE window.

   Note: This part of the script creates the Log Analytics workspace.

7. The script will display instructions to follow to authenticate to an Azure subscription. Start another instance of the Microsoft Edge based on Chromium browser, navigate to https://microsoft.com/devicelogin, and then enter the code provided in the instructions. When you receive a prompt, authenticate by using a user account with the Owner or Contributor role in the Azure subscription you are using in this lab, and then close the browser window.

   Note: If the user account is associated with multiple Azure subscriptions, the script will automatically display a grid with the list of your subscriptions. Select the one you want to use in this lab, and then select **OK**.

   Note: If you have existing Azure Log Analytics workspaces in the Azure subscription that you select, the script will automatically display a grid with the list of available Log Analytics workspaces in the Azure subscription you selected. Select **Cancel**, and the script will automatically provision one with a name that consists of the **WSLabWinAnalytics** prefix followed by the Azure subscription ID.

8. The script will automatically display a grid containing the list of Azure regions. Select **eastus**, and then select **OK**.

   Note: Make sure to select **eastus** as the target Azure region. The Azure Log Analytics location and the corresponding Azure Automation account locations must follow mappings documented in Supported regions for linked Log Analytics workspace.

9. Within the console session to the **WSLabOnboard-DC** VM, in the PowerShell ISE window, in the **C:\Library\Scenario.ps1** file in the script pane, select the fifth part of the script between the lines **107** and **151** marked as **#region setup Log Analytics Gateway**, and then run that part of the script either by selecting the **F8** key or selecting the **Run selection** button in the toolbar of the PowerShell ISE window.

> **Note**: This part of the script installs Log Analytics Gateway.
>
> **Note**: Disregard the warning about breaking changes to the cmdlet **Get-AzOperationalInsightsWorkspaceSh**

14.2.6 Task 6: Configure Azure Log Analytics workspace

1. Within the console session to the **WSLabOnboard-DC** VM, switch back to the browser window displaying the Azure portal interface.
2. In the Azure portal, in the **Search resources, services, and docs** text box at the top of the Azure portal page, enter **Log Analytics workspaces**, and then select the **Enter** key.
3. On the **Log Analytics workspaces** blade, select the workspace you created in the previous task.

   > **Note**: The workspace name has the **WSLabWorkspace** prefix.

4. On the Log Analytics workspace blade, in the **Settings** section, select **Agents Configuration**.
5. In the **Windows event logs** tab, enter **System**, and then select **+ Add windows event log**.
6. Use the procedure described in the previous step to add the **Application** log.
7. On the **Agents Configuration** blade, select **Windows performance counters** tab, select **+ Add performance counter**, enter **Processor(*)\% Processor Time** and select **Apply**.

14.2.7 Task 7: Integrate hyperconverged infrastructure with Azure Automation

1. Within the console session to the **WSLabOnboard-DC** VM, switch to the PowerShell ISE window and, in the **C:\Library\Scenario.ps1** file in the script pane, replace line **163**

   ```
   $location=(Get-AzOperationalInsightsWorkspace -Name $WorkspaceName -ResourceGroupName $ResourceGro
   ```

   with the following code:

   ```
   $location = 'eastus2'
   ```

   > **Note**: This ensures that the location of the Azure Automation account maps to the location of the Azure Log Analytics workspace, as documented in Supported regions for linked Log Analytics workspace.

2. Within the console session to the **WSLabOnboard-DC** VM, in the PowerShell ISE window, in the **C:\Library\Scenario.ps1** file in the script pane, select the sixth part of the script between the lines **153** and **287** marked as **#region deploy a Windows Hybrid Runbook Worker**, and then run that part of the script either by selecting the **F8** key or selecting the **Run selection** button in the toolbar of the PowerShell ISE window.

   > **Note**: This part of the script creates an Azure Automation Account and configures Hybrid Runbook Worker on the **HRWorker01** VM.
   >
   > **Note**: Disregard the warning about breaking changes to the cmdlet **Get-AzOperationalInsightsWorkspaceSh**
   >
   > **Note**: Disregard error messages during registration of the Hybrid Runbook Worker. You can verify that the registration was successful by switching to the browser displaying the Azure portal interface, navigating to the **WSLabAutomationAccount** Azure Automation account you created in this task, selecting **Hybrid worker groups**, and then finally selecting the **System hybrid worker groups**. You will find the `HRWorker01.Corp.contoso.com` entry there, representing the newly registered Hybrid Runbook worker.

3. Within the console session to the **WSLabOnboard-DC** VM, in the PowerShell ISE window, in the **C:\Library\Scenario.ps1** file in the script pane, replace line **286**

   ```
   $location=(Get-AzOperationalInsightsWorkspace -Name $WorkspaceName -ResourceGroupName $ResourceGro
   ```

   with the following code:

   ```
   $location = 'eastus2'
   ```

4. Within the console session to the **WSLabOnboard-DC** VM, in the PowerShell ISE window, in the **C:\Library\Scenario.ps1** file in the script pane, select the seventh part of the script between the lines **289** and **328** marked as **#region configure Hybrid Runbook Worker Addresses and Azure Automation Agent Service URL on Log Analytics Gateway**, and then run that part of the script

either by selecting the **F8** key or selecting the **Run selection** button in the toolbar of the PowerShell ISE window.

> **Note**: This part configures the Log Analytics Gateway to connect to the Azure Automation endpoints.

14.2.8 Task 8: Integrate Storage Spaces Direct cluster nodes with with Azure Monitor

1. Within the console session to the **WSLabOnboard-DC** VM, in the PowerShell ISE window, in the **F:\WSLab-master\Scripts\Scenario.ps1** file in the script pane, select the eighth part of the script between the lines **330** and **372** marked as **#region download and deploy MMA Agent to S2D cluster nodes**, and then run that part of the script either by selecting the **F8** key or selecting the **Run selection** button in the toolbar of the PowerShell ISE window.

   > **Note**: This part installs the Log Analytics agent to Storage Spaces Direct cluster nodes.

   > **Note**: Disregard the warning about breaking changes to the cmdlet **Get-AzOperationalInsightsWorkspaceSh**

2. Within the console session to the **WSLabOnboard-DC** VM, in the PowerShell ISE window, in the **F:\WSLab-master\Scripts\Scenario.ps1** file in the script pane, select the ninth part of the script between the lines **374** and **401** marked as **#region download and install dependency agent (for service map solution)**, and then run that part of the script either by selecting the **F8** key or selecting the **Run selection** button in the toolbar of the PowerShell ISE window.

   > **Note**: This part installs the Dependency agent, which provides the Service Map functionality.

14.3 Exercise 3: Reviewing Azure integration functionality

14.3.1 Task 1: Review Log Analytics functionality

1. Within the console session to the **WSLabOnboard-DC** VM, switch back to the browser window displaying the Azure portal interface.
2. In the Azure portal, in the **Search resources, services, and docs** text box at the top of the Azure portal page, enter **Log Analytics workspaces**, and then select the **Enter** key.
3. On the **Log Analytics workspaces** blade, select the workspace you created earlier in this lab (its name starts with the **WSLabWorkspace** prefix).
4. On the Log Analytics workspace blade, in the **General** section, select **Logs**, and then select **Get Started**.
5. In the **Example queries** pane, in the list of **All Queries**, select **Virtual machines**. In the list of sample queries, select **Top 10 Virtual Machines by CPU utilization in the last 7 days**, and then select **Run**. This will automatically display the corresponding query and its results.

   > **Note**: Review the query and the results.

   > **Note**: The query might result in the syntax error message if the data has not been collected yet. If so, wait for a few minutes and try again or return to this task once you complete the rest of the lab.

14.3.2 Task 2: Review Azure Automation functionality

1. Within the console session to the **WSLabOnboard-DC** VM, in the browser window displaying the Azure portal, navigate back to the blade displaying the Log Analytics workspace you were reviewing in the previous task.
2. On the Log Analytics workspace blade, in the **Related Resources** section, select **Automation Account**.
3. Note the information regarding the linked Automation account, and then select **Go to account**. You will be redirected to the **WSLabAutomationAccount** blade.
4. On the **WSLabAutomationAccount** blade, in the **Configuration Management** section, select **Inventory**.
5. On the **Inventory** blade, note that you have the option to **Enable** the Inventory solution.
6. Without making any changes, on the **WSLabAutomationAccount** blade, in the **Configuration Management** section, select **Change tracking**.

7. On the **Change tracking** blade, note that you have the option to **Enable** the Change tracking solution.
8. Without making any changes, on the **WSLabAutomationAccount** blade, in the **Process automation** section, select **Hybrid worker groups**.
9. On the **Hybrid worker groups** blade, select the **System hybrid worker groups** tab and note that it contains a separate group for each server that was registered with Azure Automation, with a single worker per group.

   > **Note**: Verify that last seen time for each worker group is within one hour of the current time.

14.3.3 Task 3: Review Service Map functionality

1. Within the console session to the **WSLabOnboard-DC** VM, in the browser window displaying the Azure portal, navigate back to the blade displaying the Log Analytics workspace you were reviewing in the first task of this exercise.
2. On the Log Analytics workspace blade, in the **General** section, select **Workspace summary**.
3. On the **Overview** blade, review the list of solutions that you implemented in the previous exercise, and then select the **Service Map** tile.

   > **Note**: It may take several minutes for the **Service Map** blade to appear.

4. On the **Service Map** blade, on the **Machines** tab, in the list of monitored servers, select **S2D1** (one of the nodes of the Storage Spaces Direct cluster), select the + icon in the diagram to the right of the server list to zoom into the diagram in the center of the blade, and then review the **Summary** pane on the right-hand side of the blade.
5. With the **S2D1** server selected, display each of the sections on the right-hand side of the pane, including **Summary**, **Properties**, **Alerts**, **Log Events**, **Performance**, **Security**, and **Updates**.

   > **Note**: In the **Security** section, if you find the **Logons with a clear text password** entry, select it. You will be automatically redirected to the Log Analytics workspace blade displaying the corresponding Kusto Query Language (KQL) query.

6. With the **S2D1** server selected, zoom in further on the diagram, and then expand the rectangle representing the **S2D2** cluster node by selecting the inverted caret character (ˇ).

   > **Note**: Review the list of connections and verify that they involve multiple processes (such as **clussvc** and **System**).

7. Review the diagram and note that it includes connections to `DC.corp.contoso.com` over ports 53 (dns), 67 (bootps), 88, 123, 135, 389, and 445 (there might be others).

   Note: These connections are listed even though the **DC** server does not have the Log Analytics and Dependency agents installed.

14.4 Exercise 4: Managing updates to hyperconverged infrastructure

14.4.1 Task 1: Implement Cluster Aware Updating by using Windows Admin Center

1. Within the console session to the **WSLabOnboard-DC** VM, switch back to the **Windows Admin Center** interface, and in the list of **Tools** of the `S2DCL1.corp.contoso.com` page, select **Updates**.
2. Select **Add Cluster-Aware Updating role**.
3. On the **Updates** panel, select **Check for updates**.
4. Review the list of available updates without making any changes.

   > **Note**: You have the option to **Apply All Updates**. Do not select it.

   > **Note**: You can monitor the status of applying the updates directly from the **Cluster Aware Updating** panel.

14.4.2 Task 2: Use Azure Automation update management

1. Within the console session to the **WSLabOnboard-DC** VM, switch back to the browser window displaying the **WSLabAutomationAccount** blade in the Azure portal.

2. On the **WSLabAutomationAccount** blade, in the **Update Management** section, select **Update Management**.
3. On the **Update Management** blade, review the list of machines, and identify noncompliant ones.

   > **Note**: To schedule an update deployment, you must first create a computer group.

4. Within the console session to the **WSLabOnboard-DC** VM, in the browser displaying the Azure portal, navigate back to the blade displaying the Log Analytics workspace you were reviewing in the previous exercise.
5. On the Log Analytics workspace blade, in the **General** section, select **Logs**.
6. On the **Example queries** pane, scroll down to the **Virtual machines** section, select it, in the listing of queries, locate the **Missing security or critical updates** from the **Virtual Machine** tile, hover over the **Run** button, and select **Load to editor**.
7. In the editor window, remove the line ’| summarize count() by Classification’, select **Run** and review results of the query.

   > **Note**: the query lists all of missing security or critical updates.

8. In the editor window, replace the query with the following one:

   ```
   Update
   | where UpdateState == 'Needed' and Optional == false and Classification == 'Security Updates' and
   | distinct Computer
   ```

   > **Note**: Computer group queries must use the `distinct Computer` clause.

   > **Note**: The query excludes servers which are members of the Storage Spaces Direct cluster because these are updated by using Cluster Aware Updating.

9. Select **Run** to verify that the query returns the list of noncompliant servers in the `Corp.contoso.com` domain that are not part of the Storage Spaces Direct cluster.
10. Select **Save**; in the drop down list, select **Save**; in the **Save** pane, specify the following settings; and then select **Save**:

    Table 2: Group query settings

Setting	Value
Name	`corp.contoso.com non-compliant non S2D servers`
Save as	Function
Function Alias	corp_non_s2d_non_compliant
Save this query as a computer group	enabled
Category	Updates

11. In the Azure portal, navigate back to the **WSLabAutomationAccount** Automation Account blade, and in the **Update Management** section, select **Update Management**.
12. On the **Update Management** blade, select **Schedule update deployment**.
13. On the **New update deployment** blade, in the **Name** text box, enter **ws01302 update deployment** and ensure that the **Operating system** switch is set to **Windows**.
14. On the **New update deployment** blade, in the **Items to update** section, select **Groups to update**.
15. On the **Select groups** blade, select the **Non-Azure** tab; in the **Available Items** section, in the `corp.contoso.com non-compliant non S2D servers` row, select **add**; and then select **OK**.
16. On the **New update deployment** blade, in the **Items to update** section, select **Machines to update**.
17. On the **Select machines** blade, select the `corp.contoso.com non-compliant non S2D servers` entry, and then select **OK**.
18. Leave the settings within the **Update classifications** drop down list with the default values.

    > **Note**: You can use the **Include/exclude updates** setting to include or exclude individual updates.

19. On the **New update deployment** blade, in the **Items to update** section, select **Schedule settings**, specify the date and time at least 5 minutes ahead of the current date and time, select your current time zone, ensure that **Recurrence** switch is set to **Once**, and then select **OK**.

    > **Note**: You can use the **Pre-scripts + Post-scripts** setting to specify scripts to run before and after patch deployment.

20. Leave the **Maintenance window (minutes)** and **Reboot options** with their default values (**120** and **Reboot if required**, respectively) and select **Create**.
21. Back on the **WSLabAutomationAccount | Update management** blade, select the **Deployment schedules** tab and ensure that the deployment has been successfully scheduled.

14.5 Exercise 5: Deprovisioning the lab environment

14.5.1 Task 1: Deprovision the Azure resources

1. Switch to the lab VM.
2. Start a browser, navigate to the Azure portal, and then sign in with the Owner or Contributor role in the Azure subscription you will be using in this lab.
3. Within the Azure portal, select the **Cloud Shell** icon in the toolbar of the portal interface.
4. If you receive a prompt to select either **Bash** or **PowerShell**, select **PowerShell**.
5. From the Cloud Shell pane, run the following to remove all Azure resources you provisioned in this lab:

```
Get-AzResourceGroup -Name 'WS013-02-RG' | Remove-AzResourceGroup -Force -AsJob
Get-AzResourceGroup -Name 'WSLabWinAnalytics' | Remove-AzResourceGroup -Force -AsJob
```

14.5.2 Task 2: Deprovision the lab resources

1. On the lab VM, start Windows PowerShell ISE as Administrator.
2. In the PowerShell ISE window, open and run the **F:\WSLab-master\Scripts\Cleanup.ps1** script to remove all VMs provisioned in this lab.

14.6 lab: title: 'Lab A: Implementing a Storage Spaces Direct cluster by using Windows PowerShell' type: 'Answer Key' module: 'Module 3: Planning for and implementing Azure Stack HCI Storage'

15 Lab A answer key: Implementing a Storage Spaces Direct cluster by using Windows PowerShell

15.1 Exercise 1: Implementing a Storage Spaces Direct cluster by using Windows PowerShell

15.1.1 Task 1: Provision the lab environment VMs

1. On the lab VM, in the **console** pane of the **Administrator: Windows PowerShell ISE** window, run the following command to rename **LabConfig.ps1** and **Scenario.ps1**:

```
Set-Location -Path 'F:\WSLab-master\Scripts'
Move-Item -Path '.\LabConfig.ps1' -Destination '.\LabConfig.m3l2.ps1' -Force -ErrorAction Silently
Move-Item -Path '.\Scenario.ps1' -Destination '.\Scenario.m3l2.ps1' -Force -ErrorAction SilentlyCo
```

2. On the lab VM, in the **Administrator: Windows PowerShell ISE** window, open a new tab on the **script** pane, paste the following command, and then save it as **F:\WSLab-master\Scripts\LabConfig.ps1**:

```
$LabConfig=@{ DomainAdminName = 'LabAdmin'; AdminPassword = 'LS1setup!'; Prefix = 'WSLab-'; Secure
1..4 | % {
     $VMNames = "S2D";
     $LABConfig.VMs += @{
     VMName = "$VMNames$_" ;
     Configuration = 'S2D' ;
```

```
      ParentVHD = 'Win2019Core_G2.vhdx';
      HDDNumber = 12;
      HDDSize = 4TB ;
      MemoryStartupBytes = 4GB;
      NestedVirt = $True
     }
}
$LabConfig.VMs += @{
      VMName = 'Management' ;
      Configuration = 'Simple';
      ParentVHD = 'Win2019_G2.vhdx';
      StaticMemory = $true;
      MemoryStartupBytes = 8GB;
      AddToolsVHD = $True;
      DisableWCF = $True;
      VMProcessorCount = 4
}
```

3. On the lab VM, in the **Administrator: Windows PowerShell ISE** window, open and run the **F:\WSLab-master\Scripts\3_Deploy.ps1** script to provision VMs for the Storage Spaces Direct environment.

    Note: For the Telemetry prompt select **None**. The script should complete in about 10 minutes. When prompted **Press enter to continue**, select the **Enter** key.

4. After the script completes, in the **Administrator: Windows PowerShell ISE** window, from the **console** pane, run the following command to start the newly provisioned VMs that will host the Storage Spaces Direct environment:

```
Get-VM -Name 'WSLab-Management' | Start-VM
Start-Sleep 150
Get-VM | Where-Object Name -like 'WSLab-S2D*' | Start-VM -AsJob
```

5. On the **console** pane of the **Administrator: Windows PowerShell ISE** window, run the following command to copy the **Scenario.ps1** file from **F:\WSLab-master\Scenarios\S2D Hyperconverged** to the current directory:

```
Copy-Item -Path 'F:\WSLab-master\Scenarios\S2D Hyperconverged\Scenario.ps1' -Destination '.\'
```

6. On the lab VM, start **Hyper-V Manager** and connect via a console session to **WSLab-DC**. When prompted to sign in, provide the username **CORP\LabAdmin** and the password **LS1setup!**.

7. In the **WSLab-DC** VM console session, start **Windows PowerShell ISE** as an administrator.

8. From the **Administrator: Windows PowerShell ISE** window, run `slmgr -rearm` and then select **OK**.

9. From the **Administrator: Windows PowerShell ISE** window, run `Restart-Computer -Force`.

    Note: Make sure that the **WSLab-DC** VM is running before you proceed to the next task.

15.1.2 Task 2: Configure the management server

1. On the lab VM, in the **Server Manager** window, select **Tools**, and then from the drop-down list, select **Hyper-V Manager**.

2. On the lab VM, in the **Hyper-V Manager** console, in the list of virtual machines, right-click or access the context menu for the **WSLab-Management** entry, and then select **Connect** to establish a console session to the **WSLab-Management** VM. When prompted to sign in, provide the username **CORP\LabAdmin** and password **LS1setup!**.

3. In the console session to the **WSLab-Management** VM, start **Windows PowerShell ISE** as Administrator.

4. In the console session to the **WSLab-Management** VM, from the **console** pane of the **Administrator: Windows PowerShell ISE** window, run the following command to install Remote Server Administration Tools:

```
Install-WindowsFeature -Name RSAT-Clustering,RSAT-Clustering-Mgmt,RSAT-Clustering-PowerShell,RSAT-
```

Note: Proceed to the next step without waiting for the installation to complete.

5. In the console session to the **WSLab-Management** VM, start another instance of Windows PowerShell ISE as Administrator.

6. In the console session to the **WSLab-Management** VM, from the **script** pane of the newly started **Administrator: Windows PowerShell ISE** window, run the following command to download and install the Microsoft Edge (Chromium) browser:

```
$progressPreference='SilentlyContinue'
Invoke-WebRequest -Uri "https://go.microsoft.com/fwlink/?linkid=2069324&language=en-us&Consent=1" -
Start-Process -FilePath "$env:USERPROFILE\Downloads\MicrosoftEdgeSetup.exe" -Wait
```

Note: Proceed to the next step without waiting for the installation to complete.

7. In the console session to the **WSLab-Management** VM, start another instance of **Windows PowerShell ISE** as Administrator.

8. In the console session to the **WSLab-Management** VM, from the **script** pane of the newly started **Administrator: Windows PowerShell ISE** window, run the following command to download and install Windows Admin Center:

```
Invoke-WebRequest -UseBasicParsing -Uri https://aka.ms/WACDownload -OutFile "$env:USERPROFILE\Downl
Start-Process msiexec.exe -Wait -ArgumentList "/i $env:USERPROFILE\Downloads\WindowsAdminCenter.ms
```

Note: Proceed to the next step without waiting for the installation to complete.

9. Return to the first **Administrator: Windows PowerShell ISE** window where you initiated installation of the Remote Server Administration Tools, wait for the installation to complete, and then from the **script** pane, run the following command to configure Kerberos constrained delegation to minimize prompts for credentials when using Windows Admin Center:

```
$gateway = "Management"
$nodes = Get-ADComputer -Filter * -SearchBase "ou=workshop,DC=corp,dc=contoso,DC=com"
$gatewayObject = Get-ADComputer -Identity $gateway
foreach ($node in $nodes){
 Set-ADComputer -Identity $node -PrincipalsAllowedToDelegateToAccount $gatewayObject
}
```

Note: Before you proceed to the next step, verify that the installation of Microsoft Edge and Windows Admin Center completed.

10. Close the other two instances of the **Administrator: Windows PowerShell ISE** window you opened earlier in this task without saving the scripts you ran from each.

11. Switch to the Microsoft Edge browser window, select **Get started**, accept the default tab page settings, select the **Continue without Signing-in** link, use the Microsoft Edge browser to navigate to `https://management.corp.contoso.com`, and when prompted to authenticate, sign in as **CORP\LabAdmin** with the password **LS1setup!**.

15.1.3 Task 3: Deploy a Storage Spaces Direct cluster on the lab VMs by using PowerShell

1. In the console session to the **WSLab-Management** VM, start File Explorer and create a **C:\Library** folder.

2. Switch to the lab VM, start File Explorer, navigate to the **F:\WSLab-master\Scripts\** folder, right-click or access the context menu for **Scenario.ps1**, and then in the context menu, select **Copy**.

3. Switch to the **WSLab-Management** VM, in the File Explorer window, right-click or access the context menu for the **C:\Library** folder, and then select **Paste**.

4. In the console session to the **WSLab-Management** VM, switch to the **Administrator: Windows PowerShell ISE** window, and in the **Administrator: Windows PowerShell ISE** window, open the **C:\Library\Scenario.ps1** script.

5. In the console session to the **WSLab-Management** VM, from the **script** pane of the **Administrator: Windows PowerShell ISE** window, run the **Scenario.ps1** script to deploy and configure a Storage Spaces Direct cluster on the lab VMs.

Note: Wait for the script to complete before you proceed to the next lab. The script should complete in about 35 minutes.

15.2 lab: title: 'Lab B: Managing storage of a Storage Spaces Direct cluster by using Windows Admin Center and Windows PowerShell' type: 'Answer Key' module: 'Module 3: Planning for and implementing Azure Stack HCI Storage'

16 Lab B answer key: Managing storage of a Storage Spaces Direct cluster by using Windows Admin Center and Windows PowerShell

16.1 Exercise 1: Managing storage of a Storage Spaces Direct cluster by using Windows Admin Center and Windows PowerShell

Note: Ensure that the **Scenario.ps1** script you started in the previous lab has completed successfully before you start this exercise.

16.1.1 Task 1: Review the installation of the Storage Spaces Direct cluster on the lab VMs

1. In the console session to the **WSLab-Management** VM, in the browser window displaying the Windows Admin Center interface, on the **All connections** page, select **+ Add**.
2. On the **Add or create resources** pane, in the **Server clusters** section, select **Add**.
3. In the **Cluster name** text box, enter `s2d-cluster.corp.contoso.com`, and then select the **Use another account for this connection** option.
4. In the **Username** text box, enter **CORP\LabAdmin**, in the **Password** text box, enter **LS1setup!**, select **Connect with account**, and then select **Add**.
5. In the console session to the **WSLab-Management** VM, in the browser window displaying the Windows Admin Center interface, on the **All connections** page, select the `s2d-cluster.corp.contoso.com` entry.
6. In the browser window displaying the Windows Admin Center interface, on the `s2d-cluster.corp.contoso.com` page, in the **Storage** section, select **Volumes**, and then on the **Volumes** pane, select the **Inventory** tab.
7. Review the list of volumes and verify that each of them is listed with a status of **OK**.
8. In the browser window displaying the Windows Admin Center interface, on the `s2d-cluster.corp.contoso.com` page, in the **Storage** section, select **Drives**.
9. On the **Drives** panel, select the **Inventory** tab, and then review the list of drives and verify that each of them is listed with a status of **OK**.
10. In the browser window displaying the Windows Admin Center interface, on the `s2d-cluster.corp.contoso.com` page, in the **Networking** section, select **Virtual switches**, on the **Virtual switches** panel, note that each node is connected to an external switch named **SETSwitch**, select one of the **SETSwitch** entries, and then select **Settings**.
11. On the **Settings for SETSwitch** panel, review the list of attached network adapters and the load balancing algorithm (set to **Hyper-V port**), and then select **Close**.
12. In the browser window displaying the Windows Admin Center interface, on the `s2d-cluster.corp.contoso.com` page, select **Settings**.
13. On the **Settings** panel, select **Storage Spaces and pools** and verify that the cluster contains a single storage pool.

    Note: You have the option to assign an arbitrary name to the storage pool.

14. On the **Storage Spaces and pools** page review the cache settings.

> **Note:** The **Cache mode for HDD** is set by default to **Read/Write** and the **Cache mode for SSD** is set to **Write only**. You have the option to modify these settings.

15. In the browser window displaying the Windows Admin Center interface, on the `s2d-cluster.corp.contoso.com` page, on the **Settings** panel, in the **Cluster** section, select the following entries:
    - Access point. Verify that **Cluster name** is set to **S2D-Cluster**.
    - Node shutdown behavior. Verify that the setting **Move virtual machines on node shutdown** is enabled.
    - Cluster traffic encryption. Verify that **Core traffic** is set to **Sign** and **Storage traffic** is set to **Clear text**.
    - Virtual machine load balancing. Verify that **Balance virtual machines** is set to **Always** with **Aggressiveness** set to **Low**.
    - Witness. Verify that **Witness type** is set to **File share witness** with **File share path** set to **\\DC\S2D-ClusterWitness**.
16. In the console session to the **WSLab-Management** VM, switch to the **Server Manager** window, select **Tools**, and in the **Tools** drop-down menu, select **Failover Cluster Manager**.
17. In the **Failover Cluster Manager** window, right-click or access the context menu for the **Failover Cluster Manager** node, and then in the context menu, select **Connect to cluster**.
18. In the **Select Cluster** dialog box, in the **Cluster name** text box, enter `s2d-cluster.corp.contoso.com`, and then select **OK**.
19. In the **Failover Cluster Manager** window, in the **Storage** node tree, select **Disks**, and then review the list of disks.
20. In the **Failover Cluster Manager** window, in the **Storage** node tree, select **Pools**, and then verify that it contains a single pool named **Cluster Pool 1**.
21. Select the **Cluster Pool 1** entry, and in the **Cluster Pool 1** pane, examine its properties by selecting the **Summary** tab, followed by the **Virtual Disks** and **Physical Disks** tabs.
22. In the **Failover Cluster Manager** window, select the **Networks** node and note that it contains separate entries for the **Management** and **SMB** networks.
23. Select the **SMB** entry, and then select the **Network connections** tab and note that it uses two network adapters on each cluster node.

16.1.2 Task 2: Create and manage volumes by using Windows Admin Center

1. In the browser window displaying the Windows Admin Center interface, on the **s2d-cluster** page, in the **Storage** section, select **Volumes**.
2. On the **Volumes** pane, select **Inventory**, and then select **Create**.

   > **Note:** At this point, the inventory includes only the pre-created **ClusterPerformanceHistory** volume.

3. On the **Create volume** pane, specify the settings listed in the following table:

   Table 1: Three-way volume settings

Setting	Value
Name	Volume01-3wm
Resiliency	Three-way mirror
Size on HDD	100
Size units	GB

4. Review the resulting estimated footprint and the total available storage space, and then select **Create** twice.
5. On the **Create volume** pane, specify the settings listed in the following table:

   Table 2: Mirror-accelerated parity volume settings

Setting	Value
Name	Volume02-map70
Resiliency	Mirror-accelerated parity
Parity percentage	70% parity, 30% mirror
Size on HDD	100
Size units	GB

6. Review the resulting estimated footprint and the total available storage space, and then select **More options**.
7. In the **More options** section, note the message indicating that to use deduplication and compression, it's necessary to install the **Data Deduplication** role on every server.
8. On the **Create volume** pane, select **Create**.
9. To install the **Data Deduplication** Windows Server role service on each cluster node, in the console session to the **WSLab-Management** VM, from the **script** pane of the **Administrator: Windows PowerShell ISE** window, run the following command:

   ```
   $servers = @('S2D1','S2D2','S2D3','S2D4')
   Invoke-Command -ComputerName $servers -ScriptBlock {Install-WindowsFeature -Name FS-Data-Deduplica
   ```

   > **Note:** Wait for the installation to complete before you proceed to the next step. This should take about three minutes.

10. Switch to the browser window displaying the Windows Admin Center interface.
11. On the `s2d-cluster.corp.contoso.com` page, on the **Volumes** panel, select the **Volume02-map70** entry, on the **Volumes > Volume Volume02-map70** pane, in the **Optional features** section, set the **Deduplication and compression** switch to **On**, and when prompted for confirmation, in the **Deduplication and compression** message box, select **Start**.
12. On the **Enable deduplication** pane, in the **Deduplication mode** drop-down list, select **Hyper-V** and then select **Enable deduplication**.

    > **Note:** You might need to close and re-open the browser page displaying the Windows Admin Center interface to account for the installation of the **Data Deduplication** role service.

13. On the **Volumes > Volume Volume02-map70** pane, select **Expand**.
14. On the **Expand volume Volume02-map70**, in the **Size on HDD (Current size 99.9 GB)** text box, enter **200**, and then select **Expand**.
15. On the **Volumes** pane, on the **Inventory** tab, in the list of volumes, select the **Volume02-map70** entry.
16. On the **Volume02-map70** pane, review the existing settings, including **Optional features**.
17. In the **Related** section, select **Storage tiers**, and verify that it contains **Dual parity** and **Three-way mirror**.

    > **Note:** You have the option to enable or disable encryption and compression, but it's not possible to modify integrity checksum or resiliency settings after the volume is created.

18. On the **Volume02-map70** pane, select **Open**.

    > **Note:** The Windows Admin Center automatically displays the page with the content of **C: > ClusterStorage > Volume02-map70** on the Hyper-V cluster node, which serves as the owner of the corresponding volume.

19. Switch to the lab VM and use the copy and paste functionality of the Hyper-V console session to copy the **tools.vhdx** file from the **F:\WSLab-master\Scripts\ParentDisks** folder to the **Downloads** folder on the **WSLab-Management** VM.
20. Switch to the console session connected to the **WSLab-Management** VM, in the browser window displaying the Windows Admin Center interface.
21. On the **C: > ClusterStorage > Volume02-map70** pane, select **New Folder**, and on the **Create New Folder** pane, in the **New folder name** text box, enter **vhdFiles**, and then select **Submit**.

22. On the **Files** pane, select the newly created folder, and on the **C: > ClusterStorage > Volume02-map70 > vhdFiles** pane, select **More**, and then select **Upload**.

23. On the **Upload** pane, select **Select files**, and in the **Open** dialog box, navigate to the **Downloads** folder, select **tools.vhdx**, and then select **Open**.

24. On the **Upload** pane, select **Submit**, and then verify that the upload completed successfully.

16.1.3 Task 3: Review the health status of the Storage Spaces Direct cluster

1. To identify the health status of the Storage Spaces Direct cluster, in the console session to the **WSLab-Management** VM, from the **script** pane of the **Administrator: Windows PowerShell ISE** window, run the following command:

   ```
   $storagesubsystem = Get-StorageSubSystem -CimSession s2d-cluster -FriendlyName Cl*
   $storagesubsystem
   ```

   Note: Ensure that **HealthStatus** is listed as **Healthy** and **OperationalStatus** as **OK**.

2. In the console session to the **WSLab-Management** VM, switch to the browser window displaying the Windows Admin Center interface and navigate to the **s2d-cluster** page.

3. On the **s2d-cluster** page, select **Dashboard** and review its contents, verifying that it reports a status of **Healthy** for all cluster components.

4. In the console session to the **WSLab-Management** VM, switch to the **Administrator: Windows PowerShell ISE** window.

5. To identify any health faults of the Storage Spaces Direct cluster, from the **console** pane of the **Administrator: Windows PowerShell ISE** window, run the following command:

   ```
   Get-HealthFault -CimSession s2d-cluster
   ```

   Note: If there are no health faults, the cmdlet should return **WARNING: Storage Spaces Direct-cluster: There aren't any faults right now**.

6. In the console session to the **WSLab-Management** VM, from the **console** pane of the **Administrator: Windows PowerShell ISE** window, run the following command to identify the health actions of the Storage Spaces Direct cluster:

   ```
   $storagesubsystem | Get-StorageHealthAction -CimSession s2d-cluster
   ```

   Note: Verify that there are no pending health actions.

7. In the console session to the **WSLab-Management** VM, from the **script** pane of the **Administrator: Windows PowerShell ISE** window, run the following command to identify the status of virtual disks of the Storage Spaces Direct cluster:

   ```
   Get-VirtualDisk -CimSession s2d-cluster | Sort-Object FriendlyName
   ```

   Note: Ensure that the **HealthStatus** is listed as **Healthy** and **OperationalStatus** as **OK**.

16.1.4 Task 4: Simulate removing a disk from the Storage Spaces Direct cluster

1. Switch to the lab VM and, from the **script** pane of the **Administrator: Windows PowerShell ISE** window, run the following command to pick a random disk in one of the nodes of the Storage Spaces Direct cluster:

   ```
   $diskToPull = Get-VM -Name WSLab-s2d* | Get-VMHardDiskDrive | Where-Object ControllerLocation -ge
   $diskToPull
   ```

2. On the lab VM, from the **script** pane of the **Administrator: Windows PowerShell ISE** window, run the following command to simulate removal of the disk you identified in the previous step:

   ```
   $pulledDiskPath = $diskToPull.Path
   $diskToPull | Remove-VMHardDiskDrive
   ```

3. Switch to the console session connected to the **WSLab-Management** VM and, from the **script** pane of the **Administrator: Windows PowerShell ISE** window, rerun the following command to identify the status of the virtual disks of the Storage Spaces Direct cluster:

   ```
   Get-VirtualDisk -CimSession s2d-cluster | Sort-Object FriendlyName
   ```

> **Note:** Verify that **HealthStatus** is listed as **Warning** and **OperationalStatus** as **Incomplete** for virtual disks.

4. In the console session to the **WSLab-Management** VM, from the **script** pane of the **Administrator: Windows PowerShell ISE** window, run the following command to identify the health status of the Storage Spaces Direct disks:

   ```
   Get-PhysicalDisk -CimSession s2d-cluster
   ```

   > **Note:** Review the output of the cmdlet and note that the **Operational Status** of one of the disks is listed as **Lost Communication**.

5. In the console session to the **WSLab-Management** VM, from the **script** pane of the **Administrator: Windows PowerShell ISE** window, rerun the following command to identify the health status of the Storage Spaces Direct cluster:

   ```
   Get-HealthFault -CimSession s2d-cluster
   ```

   > **Note:** Review the faults displayed by the Health service. You might have to wait a few minutes for the faults to display.

6. In the console session to the **WSLab-Management** VM, switch to the browser window displaying the Windows Admin Center interface, navigate back to the `s2d-cluster.corp.contoso.com` page, and in the **Storage** section, select **Volumes**, and then on the **Volumes** pane, review the content of the **Summary** tab, focusing on the list of **Alerts**.
7. On the **Volumes** pane, select the **Inventory** tab.
8. Review the list of volumes and identify the ones which are listed with the **Needs repair** status.
9. In the browser window displaying the Windows Admin Center interface, on the `s2d-cluster.corp.contoso.com` page, in the **Storage** section, select **Drives** and then on the **Drives** panel, on the **Summary** tab, review the **Alerts** section.
10. On the **Drives** panel, select the **Inventory** tab and identify the drive with the **Lost communication** status.
11. In the browser window displaying the Windows Admin Center interface, on the `s2d-cluster.corp.contoso.com` page, select **Dashboard**, and review its contents, verifying that it includes alerts indicating a drive issue.

16.1.5 Task 5: Simulate returning a disk to the Storage Spaces Direct cluster

1. Switch to the lab VM, and from the **console** pane of the **Administrator: Windows PowerShell ISE** window, run the following command to simulate returning the disk removed in the previous task back to the same node of the Storage Spaces Direct cluster:

   ```
   Add-VMHardDiskDrive -VMName $disktopull.VMName -Path $pulledDiskPath
   ```

2. Switch to the console session connected to the **WSLab-Management** VM, and from the **console** pane of the **Administrator: Windows PowerShell ISE** window, rerun the following command to identify the status of the virtual disks of the Storage Spaces Direct cluster:

   ```
   Get-VirtualDisk -CimSession s2d-cluster | Sort-Object FriendlyName
   ```

   > **Note:** The **HealthStatus** of virtual disks should be listed again as **Healthy**, with **OperationalStatus** listed as **OK**. If you observe one of the virtual disks listed as **InService**, wait for about a minute and repeat this step.

3. In the console session to the **WSLab-Management** VM, from the **console** pane of the **Administrator: Windows PowerShell ISE** window, rerun the following command to identify the health status of the Storage Spaces Direct cluster:

   ```
   Get-HealthFault -CimSession s2d-cluster
   ```

   > **Note:** The storage subsystem should return to the healthy status in about five minutes, so you might need to wait and rerun the cmdlet if you are still observing messages that indicate faults.

4. In the console session to the **WSLab-Management** VM, switch to the browser window displaying the Windows Admin Center interface.

5. On the s2d-cluster.corp.contoso.com page, in the **Storage** section, select **Volumes**, and then on the **Volumes** pane, review the content of the **Summary** tab, focusing on the list of **Alerts**.
6. On the **Volumes** pane, select the **Inventory** tab, and then review the list of volumes and verify that each of them is listed with a status of **OK**.
7. In the browser window displaying the Windows Admin Center interface, on the s2d-cluster.corp.contoso.com page, in the **Storage** section, select **Drives**.
8. On the **Drives** panel, on the **Summary** tab, verify that there are no entries in the **Alerts** section.
9. On the **Drives** panel, select the **Inventory** tab and then verify that all drives are listed with a status of **OK**.
10. In the console session to the **WSLab-Management** VM, switch to the browser window displaying the Windows Admin Center interface.
11. On the s2d-cluster.corp.contoso.com page, select **Dashboard** and review its contents, verifying that it reports a status of **Healthy** for all cluster components and a single alert indicating the sync operation.
12. In the console session to the **WSLab-Management** VM, from the **script** pane of the **Administrator: Windows PowerShell ISE** window, run the following command to identify the health status of the Storage Spaces Direct cluster:

    ```
    $storagesubsystem = Get-StorageSubSystem -CimSession s2d-cluster -FriendlyName Cl*
    $storagesubsystem
    ```

    Note: Before you proceed to the next task, verify that the **HealthStatus** is listed as **Healthy** and that the **OperationalStatus** is listed as **OK**.

16.1.6 Task 6: Simulate removing a disk and replacing it with a different one

1. Switch to the lab VM.
2. To choose a random disk from one of the nodes of the Storage Spaces Direct cluster, from the **script** pane of the **Administrator: Windows PowerShell ISE** window, run the following command:

    ```
    $diskToPull = Get-VM -Name WSLab-s2d* | Get-VMHardDiskDrive | Where-Object ControllerLocation -ge
    $diskToPull
    ```

3. To simulate removal of the disk you identified in the previous step, in the lab VM, from the **script** pane of the **Administrator: Windows PowerShell ISE** window, run the following command:

    ```
    $pulledDiskPath = $diskToPull.Path
    $diskToPull | Remove-VMHardDiskDrive
    ```

4. To simulate replacing the removed disk with another one, in the lab VM, from the **script** pane of the **Administrator: Windows PowerShell ISE** window, run the following command:

    ```
    $newDiskPath = "$(($pulledDiskPath).Substring(0,$pulledDiskPath.Length-5))_NEW.vhdx"
    New-VHD -Path $newDiskPath -SizeBytes 4TB
    Add-VMHardDiskDrive -VMName $diskToPull.VMName -Path $newDiskPath
    ```

5. Switch to the console session connected to the **WSLab-Management** VM.
6. To identify the status of the virtual disks of the Storage Spaces Direct cluster, from the **console** pane of the **Administrator: Windows PowerShell ISE** window, rerun the following command:

    ```
    Get-VirtualDisk -CimSession s2d-cluster | Sort-Object FriendlyName
    ```

    Note: Verify that the **OperationalStatus** for virtual disks is listed as **Incomplete** and the **HealthStatus** is listed as **Warning**.

7. To verify that the repair and regeneration jobs are in progress on the Storage Spaces Direct cluster, in the console session to the **WSLab-Management** VM, from the **script** pane of the **Administrator: Windows PowerShell ISE** window, rerun the following command:

    ```
    $storagesubsystem = Get-StorageSubSystem -CimSession s2d-cluster -FriendlyName Cl*
    $storagesubsystem | Get-StorageJob
    ```

    Note: If you don't observe any jobs, wait one minute and then rerun the cmdlets.

8. To identify the health status of the Storage Spaces Direct disks, in the console session to the **WSLab-Management** VM, from the **script** pane of the **Administrator: Windows PowerShell ISE** window, run the following command:

   ~~~~
   ```powershell
   Get-PhysicalDisk -CimSession s2d-cluster
   ```
   ~~~~

   > **Note:** Review the output of the cmdlet and note that **Operational Status** of one of the disks is listed as **{Removing From Pool, Lost Communication}** and **Usage** is listed as **Retired**.

9. To identify the retired disk, in the console session to the **WSLab-Management** VM, from the **console** pane of the **Administrator: Windows PowerShell ISE** window, run the following command:

   ```
   Get-PhysicalDisk -CimSession s2d-cluster -Usage retired
   ```

10. To identify the health status of the Storage Spaces Direct cluster, in the console session to the **WSLab-Management** VM, from the **script** pane of the **Administrator: Windows PowerShell ISE** window, run the following command:

    ```
    Get-HealthFault -CimSession s2d-cluster
    ```

11. Review the output of the cmdlet and note that the drive will be automatically retired after 15 minutes of lost communication.

12. In the console session to the **WSLab-Management** VM, switch to the browser window displaying the Windows Admin Center interface, on the `s2d-cluster.corp.contoso.com` page, in the **Storage** section, select **Volumes**, and on the **Volumes** pane, select the **Inventory** tab.

13. Review the list of volumes and verify that they are online but listed with a status of **Needs repair**.

14. In the browser window displaying the Windows Admin Center interface, on the `s2d-cluster.corp.contoso.com` page, in the **Storage** section, select **Drives**, and on the **Drives** panel, on the **Summary** tab, review the entry in the **Alerts** section.

15. On the **Drives** panel, select the **Inventory** tab and identify the drive listed with the status of **Retired, Removing from pool, Lost communication**.

16. To verify whether the retired disk has been automatically removed from the Storage Spaces Direct cluster, in the console session to the **WSLab-Management** VM, switch to the **Administrator: Windows PowerShell ISE** window, and then from the **console** pane, run the following command:

    ```
    Get-PhysicalDisk -CimSession s2d-cluster -Usage retired
    ```

    > **Note:** Verify that the command does not return any output. If that's not the case, wait a few minutes and rerun the command.

17. In the console session to the **WSLab-Management** VM, switch to the browser window displaying the Windows Admin Center interface.

18. On the `s2d-cluster.corp.contoso.com` page, in the **Storage** section, select **Volumes**, and then on the **Summary** tab of the **Volumes** pane, verify that all volumes are healthy.

19. On the **Volumes** pane, select the **Inventory** tab and then review the list of volumes, verifying that each of them is listed with a status of **OK**.

20. In the browser window displaying the Windows Admin Center interface, on the `s2d-cluster.corp.contoso.com` page, in the **Storage** section, select **Drives**, and then on the **Drives** panel, on the **Summary** tab, verify that there are no entries in the **Alerts** section.

21. On the **Drives** panel, select the **Inventory** tab and verify that all drives are listed with a status of **OK**.

22. In the browser window displaying the Windows Admin Center interface, on the `s2d-cluster.corp.contoso.com` page, select **Dashboard**, and review its contents, verifying that it reports a status of **Healthy** for all cluster components, and that there are no alerts listed.

16.1.7 Task 7: Deprovision the lab resources

1. Switch to the lab VM, and in the **Administrator: Windows PowerShell ISE** window, open and run the **F:\WSLab-master\Scripts\Cleanup.ps1** script to remove all VMs provisioned in this lab. When prompted, on the **console** pane, enter **Y**, and then select the **Enter** key.

2. After the script completes, select any key.
3. In the **Administrator: Windows PowerShell ISE** window, close the tab displaying the **F:\WSLab-master\Scripts\Cleanup.ps1** script.

16.2 lab: title: 'Lab C: Managing and monitoring resiliency of a Storage Spaces Direct cluster' type: 'Answer Key' module: 'Module 3: Planning for and implementing Azure Stack HCI Storage'

17 Lab C answer key: Managing and monitoring resiliency of a Storage Spaces Direct cluster

17.1 Exercise 1: Managing and monitoring resiliency of a Storage Spaces Direct cluster

17.1.1 Task 1: Provision the lab environment VMs

1. On the lab VM, from the **script** pane of the **Administrator: Windows PowerShell ISE** window, run the following command to rename **LabConfig.ps1** and **Scenario.ps1**:

```
Set-Location -Path 'F:\WSLab-master\Scripts'
Move-Item -Path '.\LabConfig.ps1' -Destination '.\LabConfig.m3l3.ps1' -Force -ErrorAction SilentlyC
Move-Item -Path '.\Scenario.ps1' -Destination '.\Scenario.m3l3.ps1' -Force -ErrorAction SilentlyCon
```

2. In the **Administrator: Windows PowerShell ISE** window, open a new tab on the **script** pane, paste the following command, and save it as **F:\WSLab-master\Scripts\LabConfig.ps1**:

```
$LabConfig=@{ DomainAdminName='LabAdmin'; AdminPassword='LS1setup!'; Prefix = 'WSLab-'; SwitchName
1..6 | % {
     $VMNames="S2D";
     $LABConfig.VMs += @{
     VMName = "$VMNames$_" ;
     Configuration = 'S2D' ;
     ParentVHD = 'Win2019Core_G2.vhdx';
     SSDNumber = 0;
     SSDSize=800GB ;
     HDDNumber = 12;
     HDDSize= 4TB ;
     MemoryStartupBytes= 1GB;
     NestedVirt=$false
    }
}
$LabConfig.VMs += @{
     VMName = 'Management' ;
     Configuration = 'Simple';
     ParentVHD = 'Win2019_G2.vhdx';
     StaticMemory = $true;
     MemoryStartupBytes = 8GB;
     AddToolsVHD = $True;
     DisableWCF = $True;
     VMProcessorCount = 4
}
```

3. On the lab VM, in the **Administrator: Windows PowerShell ISE** window, open and run the **F:\WSLab-master\Scripts\3_Deploy.ps1** script to provision VMs for the Storage Spaces Direct environment.

    Note: Select **None** at the Telemetry prompt. The script should complete in about 10 minutes. When prompted **Press enter to continue**, select the **Enter** key.

4. When the script completes, in the **Administrator: Windows PowerShell ISE** window, open a new tab, and run the following command to start the newly provisioned VMs that will host the Storage Spaces Direct environment:

```
Get-VM -Name 'WSLab-Management' | Start-VM
Start-Sleep 150
Get-VM | Where-Object Name -like 'WSLab-S2D*' | Start-VM -AsJob
```

5. On the lab VM, start **Hyper-V Manager** and connect via a console session to **WSLab-DC**. When prompted to sign in, provide the username **CORP\LabAdmin** and the password **LS1setup!**.
6. In the **WSLab-DC** VM console session, start **Windows PowerShell ISE** as an administrator.
7. From the **Administrator: Windows PowerShell ISE** window, run `slmgr -rearm` and then select **OK**.
8. From the **Administrator: Windows PowerShell ISE** window, run `Restart-Computer -Force`.

   Note: Make sure that the **WSLab-DC** VM is running before you proceed to the next task.

17.1.2 Task 2: Configure the management server

1. On the lab VM, in the **Server Manager** window, select **Tools**, and then in the drop-down list, select **Hyper-V Manager**.
2. On the lab VM, in the **Hyper-V Manager** console, in the list of virtual machines, right-click or access the context menu for **WSLab-Management** entry, and then select **Connect** to establish a console session to the **WSLab-Management** VM. When prompted to sign in, provide the **CORP\LabAdmin** username and **LS1setup!** password.
3. In the console session to the **WSLab-Management** VM, start Windows PowerShell ISE as Administrator.
4. In the console session to the **WSLab-Management** VM, from the **console** pane of the **Administrator: Windows PowerShell ISE** window, run the following command to install Remote Server Administration Tools:

   ```
   Install-WindowsFeature -Name RSAT-Clustering,RSAT-Clustering-Mgmt,RSAT-Clustering-PowerShell,RSAT-
   ```

   Note: Proceed to the next step without waiting for the installation to complete.
5. In the console session to the **WSLab-Management** VM, start another instance of Windows PowerShell ISE as Administrator.
6. In the console session to the **WSLab-Management** VM, from the **script** pane of the newly started **Administrator: Windows PowerShell ISE** window, run the following command to download and install the Microsoft Edge (Chromium) browser:

   ```
   $progressPreference='SilentlyContinue'
   Invoke-WebRequest -Uri "https://go.microsoft.com/fwlink/?linkid=2069324&language=en-us&Consent=1"
   Start-Process -FilePath "$env:USERPROFILE\Downloads\MicrosoftEdgeSetup.exe" -Wait
   ```

   Note: Proceed to the next step without waiting for the installation to complete.
7. In the console session to the **WSLab-Management** VM, start another instance of Windows PowerShell ISE as Administrator.
8. In the console session to the **WSLab-Management** VM, from the **script** pane of the newly started **Administrator: Windows PowerShell ISE** window, run the following command to download and install Windows Admin Center:

   ```
   Invoke-WebRequest -UseBasicParsing -Uri https://aka.ms/WACDownload -OutFile "$env:USERPROFILE\Downl
   Start-Process msiexec.exe -Wait -ArgumentList "/i $env:USERPROFILE\Downloads\WindowsAdminCenter.ms
   ```

   Note: Proceed to the next step without waiting for the installation to complete.
9. Switch to the first **Administrator: Windows PowerShell ISE** window where you initiated installation of the Remote Server Administration Tools and wait for the installation to complete.
10. To configure Kerberos constrained delegation to minimize prompts for credentials when using Windows Admin Center, from the **script** pane, run the following command:

    ```
    $gateway = "Management"
    $nodes = Get-ADComputer -Filter * -SearchBase "ou=workshop,DC=corp,dc=contoso,DC=com"
    $gatewayObject = Get-ADComputer -Identity $gateway
    foreach ($node in $nodes){
    ```

```
 Set-ADComputer -Identity $node -PrincipalsAllowedToDelegateToAccount $gatewayObject
}
```

> **Note:** Before you proceed to the next step, verify that the installation of Microsoft Edge and Windows Admin Center completed.

11. Close the other two instances of the **Administrator: Windows PowerShell ISE** window you opened earlier in this task without saving the scripts you ran from each.
12. Switch to the Microsoft Edge browser window, select **Get started**, accept the default tab page settings, and select the **Continue without Signing-in** link, use the Microsoft Edge browser to navigate to `https://management.corp.contoso.com`, and then when prompted to authenticate, sign in as **CORP\LabAdmin** with the password **LS1setup!**.

17.1.3 Task 3: Create and configure a failover cluster

1. To provision a failover cluster, in the console session to the **WSLab-Management** VM, from the **Administrator: Windows PowerShell ISE** window, run the following command:

```
$servers = 1..6 | % {"S2D$_"}
$clusterName = "S2D-Cluster"
$clusterIP = "10.0.0.111"

# Install features on servers
Invoke-Command -computername $servers -ScriptBlock {
    Install-WindowsFeature -Name "Failover-Clustering","Hyper-V-PowerShell","RSAT-Clustering-PowerS
}

# Restart servers since failover clustering in Windows Server 2019 requires reboot
Restart-Computer -ComputerName $servers -Protocol WSMan -Wait -For PowerShell

# Create cluster
New-Cluster -Name $clusterName -Node $servers -StaticAddress $clusterIP
Start-Sleep 5
Clear-DNSClientCache

# Add File Share Witness
# Create a new directory
$witnessName = $clusterName+"Witness"
Invoke-Command -ComputerName DC -ScriptBlock {New-Item -Path c:\Shares -Name $using:WitnessName -I
$accounts = @()
$accounts += "CORP\$($clusterName)$"
$accounts += "CORP\Domain Admins"
New-SmbShare -Name $witnessName -Path "c:\Shares\$witnessName" -FullAccess $accounts -CimSession D
# Set NTFS permissions
Invoke-Command -ComputerName DC -ScriptBlock {(Get-SmbShare $using:witnessName).PresetPathAcl | Se
# Set Quorum
Set-ClusterQuorum -Cluster $clusterName -FileShareWitness "\\DC\$witnessName"
```

> **Note:** Wait until the script completes before you proceed to the next task. This might take about 10 minutes.

17.1.4 Task 4: Configure fault domains on the failover cluster

1. To configure fault domains on the Storage Spaces Direct cluster, in the console session to the **WSLab-Management** VM, from the **Administrator: Windows PowerShell ISE** window, run the following command:

```
$clusterName = "S2D-Cluster"

# Create Fault domains with PowerShell
New-ClusterFaultDomain -Name "Rack01" -FaultDomainType Rack -Location "Contoso HQ, Room 4010, Aisl
New-ClusterFaultDomain -Name "Rack02" -FaultDomainType Rack -Location "Contoso HQ, Room 4010, Aisl
New-ClusterFaultDomain -Name "Rack03" -FaultDomainType Rack -Location "Contoso HQ, Room 4010, Aisl
```

```
# Assign fault domains
# Assign nodes to racks
1..2 |ForEach-Object {Set-ClusterFaultDomain -Name "S2D$_" -Parent "Rack01" -CimSession $clusterNa
3..4 |ForEach-Object {Set-ClusterFaultDomain -Name "S2D$_" -Parent "Rack02" -CimSession $clusterNa
5..6 |ForEach-Object {Set-ClusterFaultDomain -Name "S2D$_" -Parent "Rack03" -CimSession $clusterNa
```

2. To display the newly configured fault domains, in the console session to the **WSLab-Management** VM, from the **Administrator: Windows PowerShell ISE** window, run the following command:

```
$clusterName = "S2D-Cluster"
Get-ClusterFaultDomain -CimSession $clusterName
Get-ClusterFaultDomainxml -CimSession $clusterName
```

3. In the console session to the **WSLab-Management** VM, switch to the browser window displaying the Windows Admin Center interface, navigate to the **All connections** page, and then select **+ Add**.
4. On the **Add or create resources** panel, on the **Server clusters** tile, select **Add**.
5. In the **Cluster name** text box, enter `s2d-cluster.corp.contoso.com`. If prompted, select the **Use another account for this connection** option.
6. In the **Username** text box, enter **CORP\LabAdmin**, in the **Password** text box, enter **LS1setup!**, select **Connect with account**, and then select **Add**.
7. In the browser window displaying the Windows Admin Center interface, on the `s2d-cluster.corp.contoso.com` page, in the **Compute** section, select **Nodes**, and on the **Nodes** pane, review the rack information for each node.

   > **Note:** If necessary, select **Install** to install **RSAT-Clustering-PowerShell**, which is required by Windows Admin Center.

17.1.5 Task 5: Enable Storage Spaces Direct on the failover cluster

1. To enable Storage Spaces Direct on the cluster, in the console session to the **WSLab-Management** VM, from the **script** pane of the **Administrator: Windows PowerShell ISE** window, run the following command, selecting **Yes to All** when prompted for confirmation:

```
$clusterName = "S2D-Cluster"
Enable-ClusterS2D -CimSession $clusterName -Verbose
```

   > **Note:** Wait for the installation to complete before you proceed to the next task. This should take about three minutes.

2. Review the provisioning steps and note that the Storage Spaces Direct cluster setup has automatically set the default fault domain awareness on the clustered storage subsystem.

17.1.6 Task 6: Review fault domain configuration on the Storage Spaces Direct cluster

1. In the console session to the **WSLab-Management** VM, from the **script** pane of the **Administrator: Windows PowerShell ISE** window, run the following command to list storage pool properties:

```
$clusterName = "S2D-Cluster"
Get-StoragePool -CimSession $clusterName -FriendlyName S2D* | fl *
```

   > **Note:** Verify that **FaultDomainAwarenessDefault** is automatically set to **StorageRack**.

2. In the console session to the **WSLab-Management** VM, from the **console** pane of the **Administrator: Windows PowerShell ISE** window, run the following command to list properties of storage tiers:

```
Get-StorageTier -CimSession s2d-cluster | fl *
```

   > **Note:** Verify that the two tiers named **MirrorOnHDD** and **Capacity** have **FaultDomainAwarenessDefault** set to **StorageRack**.

17.1.7 Task 7: Create tiered volumes on a Storage Spaces Direct failover cluster

1. In the console session to the **WSLab-Management** VM, from the **console** pane of the **Administrator: Windows PowerShell ISE** window, run the following command to create a volume referencing the **Capacity** tier:

```
New-Volume -StoragePoolFriendlyName s2d* -FriendlyName WithTier -FileSystem CSVFS_ReFS -StorageTie
```

2. In the console session to the **WSLab-Management** VM, from the **console** pane of the **Administrator: Windows PowerShell ISE** window, run the following command to create a volume not referencing any specific tier:

   ```
   New-Volume -StoragePoolFriendlyName s2d* -FriendlyName WithoutTier -FileSystem CSVFS_ReFS -Size 1T
   ```

3. In the console session to the **WSLab-Management** VM, refresh the browser window displaying the Windows Admin Center interface.
4. On the `s2d-cluster.corp.contoso.com` page, in the **Storage** section, select **Volumes**, and then select **Inventory**.
5. On the **Volumes** pane, select the **WithoutTier** volume entry.
6. On the **Volumes > Volume WithoutTier** pane, note that **Fault domain awareness** is set to **Rack**.

   > **Note:** The actual **FaultDomainAwareness** property is defined on the virtual disk level.

7. Navigate back to the **Volumes** pane, and on the **Volumes** pane, select the **WithTier** volume entry.
8. On the **Volumes > Volume WithTier** pane, note that **Fault domain awareness** is also set to **Rack**.

   > **Note:** The actual **FaultDomainAwareness** property is defined on the storage tier associated with the tiered disk.

17.1.8 Task 8: Test resiliency of the Storage Spaces Direct cluster

1. Switch to the lab VM.
2. In the **console** pane of the **Administrator: Windows PowerShell ISE** window, run the following command to simulate a failure of an entire rack containing two cluster nodes:

   ```
   Get-VM -Name "WSLab-S2D1","WSLab-S2D2" | Stop-VM -TurnOff
   ```

3. Switch to the console session connected to the **WSLab-Management** VM.
4. In the browser window displaying the Windows Admin Center interface, on the `s2d-cluster.corp.contoso.com` page, in the **Compute** section, select **Servers**.
5. On the **Servers** pane, select the **Inventory** tab, and verify that the **S2D1** and **S2D2** nodes in **Rack01** are down but the cluster remains online.

   > **Note:** You might need to refresh the page displaying the Windows Admin Center interface to observe the updated status of cluster nodes.

6. In the console session to the **WSLab-Management** VM, in the browser window displaying the Windows Admin Center interface, on the `s2d-cluster.corp.contoso.com` page, in the **Storage** section, select **Volumes**, and on the **Volumes** pane, verify that all volumes are healthy.
7. Switch to the lab VM, and in the **console** pane of the **Administrator: Windows PowerShell ISE** window, run the following command to simulate a storage failure caused by removing one capacity disk from both **S2D3** and **S2D4** cluster nodes:

   ```
   Get-VM -Name "WSLab-S2D3","WSLab-S2D4" | Get-VMHardDiskDrive | Where-Object controllerlocation -eq
   ```

8. Switch to the console session connected to the **WSLab-Management** VM, and in the browser window displaying the Windows Admin Center interface, on the `s2d-cluster.corp.contoso.com` page, on the **Volumes** pane, verify that all volumes are still healthy.
9. In the console session to the **WSLab-Management** VM, in the browser window displaying the Windows Admin Center interface, on the `s2d-cluster.corp.contoso.com` page, in the **Storage** section, select **Drives**, and on the **Drives** pane, on the **Summary** tab, verify that **26** out of **72** drives are listed as **Critical**.
10. Switch to the lab VM, and on the **console** pane of the **Administrator: Windows PowerShell ISE** window, run the following command to simulate a failure of all disks attached to both **S2D3** and **S2D4** cluster nodes:

    ```
    Get-VM -Name "WSLab-S2D3","WSLab-S2D4" | Get-VMHardDiskDrive | Where-Object controllerlocation -ne
    ```

11. Switch to the console session connected to the **WSLab-Management** VM.

12. In the browser window displaying the Windows Admin Center interface, on the `s2d-cluster.corp.contoso.com` page, in the **Storage** section, on the **Drives** pane, on the **Summary** tab, verify that **48** out of **72** drives are listed as **Critical**.

13. In the console session to the **WSLab-Management** VM, in the browser window displaying the Windows Admin Center interface, on the `s2d-cluster.corp.contoso.com` page, in the **Storage** section, select **Volumes**, and then on the **Volumes** pane, verify that all volumes are healthy.

    > **Note:** The storage pool with all volumes might transition to the offline state if you trigger subsequent faults too quickly. If this happens, consider repeating this exercise and pausing between the steps of this task.

    > **Note:** The cluster, pool, and virtual disks are all capable of surviving another node failure if their respective resources reside in the surviving rack.

14. In the console session to the **WSLab-Management** VM, switch to the Server Manager window, select **Tools**, and then in the **Tools** drop-down menu, select **Failover Cluster Manager**.

15. In the **Failover Cluster Manager** window, right-click or access the context menu for the **Failover Cluster Manager** node, and then in the context menu, select **Connect to cluster**.

16. In the **Select Cluster** dialog box, in the **Cluster name** text box, enter `s2d-cluster.corp.contoso.com`, and then select **OK**.

17. In the console session to the **WSLab-Management** VM, in the **Failover Cluster Manager** window, navigate to the **Disks** subnode of the **Storage** node, and then identify the owner node of the **With Tier** and **Without Tier** virtual disks.

    1. If the owner node is listed as **S2D3** or **S2D4**, right-click or access the context menu for the virtual disk, and then in the context menu, select **Move**, followed by **Select Node**. In the **Move Cluster Shared Volume** dialog box, select **S2D5** or **S2D6**, and then select **OK**.

18. In the console session to the **WSLab-Management** VM, in the **Failover Cluster Manager** window, in the **Storage** node, select the **Pools** subnode, and identify the owner node of the **Cluster Pool 1** storage pool.

    1. If the owner node is listed as **S2D3** or **S2D4**, right-click or access the context menu for **Cluster Pool 1**, and then in the context menu, select **Move**, followed by **Select Node**. In the **Move Cluster Shared Volume** dialog box, select **S2D5** or **S2D6**, and then select **OK**.

19. Switch to the lab VM, and in the **console** pane of the **Administrator: Windows PowerShell ISE** window, run the following command to simulate a **S2D3** node failure:

    ```
    Stop-VM -Name "WSLab-S2D3" -TurnOff
    ```

20. Switch to the console session connected to the **WSLab-Management** VM.

21. In the console session to the **WSLab-Management** VM, in the **Failover Cluster Manager** window, select **Nodes**, and then verify that **S2D1**, **S2D2**, and **S2D3** nodes are down but the cluster remains online.

22. In the console session to the **WSLab-Management** VM, in the **Failover Cluster Manager** window, in the **Storage** node, select the **Disks** subnode, and verify that all virtual disks are online.

    > **Note:** The disks might transition to the offline state if you trigger subsequent faults too quickly.

23. In the console session to the **WSLab-Management** VM, switch to the browser window displaying the Windows Admin Center interface.

24. On the `s2d-cluster.corp.contoso.com` page, in the **Storage** section, on the **Volumes** pane, verify that all volumes are healthy.

25. In the browser window displaying the Windows Admin Center interface, on the `s2d-cluster.corp.contoso.com` page, in the **Storage** section, select **Drives**, and on the **Drives** pane, on the **Summary** tab, verify that **48** out of **72** drives are listed as **Critical**.

26. Review the results and verify that the cluster and its virtual disks remain online with only **24** out of **72** physical disks.

> **Note:** The storage pool with all volumes might transition to the offline state if you trigger subsequent faults too quickly. If this happens, consider repeating this exercise and pausing between the steps of this task.

17.1.9 Task 9: Restore failed disks and nodes on the Storage Spaces Direct cluster

1. Switch to the lab VM, and from the **script** pane of the **Administrator: Windows PowerShell ISE** window, run the following command to return all disks to the **S2D3** and **S2D4** cluster nodes:

```
$vmNames = "WSLab-S2D3","WSLab-S2D4"
foreach ($vmName in $vmNames){
  $vhds = (Get-ChildItem -Path "$((get-vm $vmName).ConfigurationLocation)\Virtual Hard Disks" | Whe
  foreach ($vhd in $vhds){
     Add-VMHardDiskDrive -VMName $VMName -Path $VHD
  }
}
```

2. On the lab VM, from the **console** pane of the **Administrator: Windows PowerShell ISE** window, run the following command to start the **S2D1**, **S2D2**, and **S2D3** cluster nodes:

```
Start-VM -Name "WSLab-S2D1","WSLab-S2D2","WSLab-S2D3"
```

3. Switch to the console session connected to the **WSLab-Management** VM, and then in the console session to the **WSLab-Management** VM, from the **console** pane of the **Administrator: Windows PowerShell ISE** window, run the following command to identify the repair and regeneration jobs in progress:

```
Get-StorageSubSystem -CimSession s2d-cluster -FriendlyName CL* | Get-StorageJob
```

4. In the console session to the **WSLab-Management** VM, switch to the browser window displaying the Windows Admin Center interface.
5. On the `s2d-cluster.corp.contoso.com` page, in the **Storage** section, on the **Drives** pane, select the **Inventory** tab, and then review the status of the drives.
6. In the browser window displaying the Windows Admin Center interface, on the `s2d-cluster.corp.contoso.com` page, select **Dashboard** and review its contents, verifying that it reports a status of **Healthy** for all cluster components with no alerts listed.

> **Note:** The storage subsystem should return to the healthy status in about five minutes, so you might need to wait and rerun the cmdlet if you are still observing messages indicating storage faults.

7. In the console session connected to the **WSLab-Management** VM, switch to the **Administrator: Windows PowerShell ISE** window and, from the **console** pane, run the following command to identify the health status of the Storage Spaces Direct cluster:

```
Get-HealthFault -CimSession s2d-cluster
```

> **Note:** Ignore faults regarding memory consumption on cluster nodes; these are expected.

17.1.10 Task 10: Deprovision the lab resources

1. Switch to the lab VM, and in the **Administrator: Windows PowerShell ISE** window, open and run the **F:\WSLab-master\Scripts\Cleanup.ps1** script to remove all VMs provisioned in this lab. When prompted, on the **console** pane, enter **Y**, and then select **Enter**.
2. When the script completes, select any key.
3. In the **Administrator: Windows PowerShell ISE** window, close the tab displaying the **F:\WSLab-master\Scripts\Cleanup.ps1** script.

17.2 lab: title: 'Lab D: Managing Storage Spaces Direct cluster tiers' type: 'Answer Key' module: 'Module 3: Planning for and implementing Azure Stack HCI Storage'

18 Lab D answer key: Managing Storage Spaces Direct cluster tiers

18.1 Exercise 1: Managing Storage Spaces Direct cluster tiers

18.1.1 Task 1: Provision the lab environment VMs

1. On the lab VM, from the **script** pane of the **Administrator: Windows PowerShell ISE** window, run the following command to rename **LabConfig.ps1** and **Scenario.ps1**:

   ```
   Set-Location -Path 'F:\WSLab-master\Scripts'
   Move-Item -Path '.\LabConfig.ps1' -Destination '.\LabConfig.m3l4.ps1' -Force -ErrorAction SilentlyC
   Move-Item -Path '.\Scenario.ps1' -Destination '.\Scenario.m3l4.ps1' -Force -ErrorAction SilentlyCon
   ```

2. In the **Administrator: Windows PowerShell ISE** window, open a new tab in the **script** pane, paste the following command, and then save it as **F:\WSLab-master\Scripts\LabConfig.ps1**:

   ```
   $LabConfig=@{ DomainAdminName = 'LabAdmin'; AdminPassword = 'LS1setup!'; Prefix = 'WSLab-'; Switch
   1..2 | % {$VMNames = "2T2node"; $LABConfig.VMs += @{ VMName = "$VMNames$_" ; Configuration = 'S2D'
   1..3 | % {$VMNames = "2T3node"; $LABConfig.VMs += @{ VMName = "$VMNames$_" ; Configuration = 'S2D'
   1..2 | % {$VMNames = "3T2node"; $LABConfig.VMs += @{ VMName = "$VMNames$_" ; Configuration = 'S2D'
   1..3 | % {$VMNames = "3T3node"; $LABConfig.VMs += @{ VMName = "$VMNames$_" ; Configuration = 'S2D'

   $LABConfig.VMs += @{
        VMName = "Management" ;
        Configuration = 'S2D' ;
        ParentVHD = 'Win2019_G2.vhdx';
        SSDNumber = 1;
        SSDSize = 50GB ;
        MemoryStartupBytes= 8GB;
        NestedVirt = $false;
        StaticMemory = $true;
        VMProcessorCount = 4
    }
   ```

3. On the lab VM, in the **Administrator: Windows PowerShell ISE** window, open and run the **F:\WSLab-master\Scripts\3_Deploy.ps1** script to provision VMs for the Storage Spaces Direct environment.

   > **Note:** Select **None** at the telemetry prompt. The script should complete in about 15 minutes. When prompted to **Press enter to continue**, select the **Enter** key.

4. After the script completes, in the **Administrator: Windows PowerShell ISE** window, from the **console** pane, run the following command to start the newly provisioned VMs that will host the Storage Spaces Direct environment:

   ```
   Get-VM -Name 'WSLab-Management' | Start-VM
   Start-Sleep 150
   Get-VM | Where-Object Name -like 'WSLab-*node*' | Start-VM -AsJob
   ```

5. On the lab VM, start **Hyper-V Manager** and connect via a console session to **WSLab-DC**. When prompted to sign in, provide the username **CORP\LabAdmin** and the password **LS1setup!**.

6. In the **WSLab-DC** VM console session, start **Windows PowerShell ISE** as an administrator.

7. From the **Administrator: Windows PowerShell ISE** window, run `slmgr -rearm` and then select **OK**.

8. From the **Administrator: Windows PowerShell ISE** window, run `Restart-Computer -Force`.

   Note: Make sure that the **WSLab-DC** VM is running before you proceed to the next task.

18.1.2 Task 2: Configure the management server

1. On the lab VM, in the **Server Manager** window, select **Tools**, and then from the drop-down list, select **Hyper-V Manager**.

2. On the lab VM, in the **Hyper-V Manager** console, in the list of virtual machines, right-click or access the context menu for the **WSLab-Management** entry, and select **Connect** to establish a console session to the **WSLab-Management** VM. When prompted to sign in, provide the **CORP\LabAdmin** username and **LS1setup!** password.

3. In the console session to the **WSLab-Management** VM, start Windows PowerShell ISE as Administrator.

4. In the console session to the **WSLab-Management** VM, from the **console** pane of the **Administrator: Windows PowerShell ISE** window, run the following command to install Remote Server Administration Tools:

```
Install-WindowsFeature -Name RSAT-Clustering,RSAT-Clustering-Mgmt,RSAT-Clustering-PowerShell,RSAT-
```

   > **Note:** Proceed to the next step without waiting for the installation to complete.

5. In the console session to the **WSLab-Management** VM, start another instance of Windows PowerShell ISE as Administrator.

6. In the console session to the **WSLab-Management** VM, from the **script** pane of the newly started **Administrator: Windows PowerShell ISE** window, run the following command to download and install Windows Admin Center:

```
Invoke-WebRequest -UseBasicParsing -Uri https://aka.ms/WACDownload -OutFile "$env:USERPROFILE\Downl
Start-Process msiexec.exe -Wait -ArgumentList "/i $env:USERPROFILE\Downloads\WindowsAdminCenter.ms
```

   > **Note:** Proceed to the next step without waiting for the installation to complete.

7. In the console session to the **WSLab-Management** VM, from the **script** pane of the newly started **Administrator: Windows PowerShell ISE** window, run the following command to download and install the Microsoft Edge (Chromium) browser:

```
$progressPreference='SilentlyContinue'
Invoke-WebRequest -Uri "https://go.microsoft.com/fwlink/?linkid=2069324&language=en-us&Consent=1" -
Start-Process -FilePath "$env:USERPROFILE\Downloads\MicrosoftEdgeSetup.exe" -Wait
```

   > **Note:** Proceed to the next step without waiting for the installation to complete.

8. To configure Kerberos constrained delegation to minimize prompts for credentials when using Windows Admin Center, switch to the first **Administrator: Windows PowerShell ISE** window where you initiated installation of the Remote Server Administration Tools, wait for the installation to complete, and then from the **script** pane, run the following command:

```
$gateway = "Management"
$nodes = Get-ADComputer -Filter * -SearchBase "ou=workshop,DC=corp,dc=contoso,DC=com"
$gatewayObject = Get-ADComputer -Identity $gateway
foreach ($node in $nodes){
 Set-ADComputer -Identity $node -PrincipalsAllowedToDelegateToAccount $gatewayObject
}
```

   > **Note:** Before you proceed to the next step, verify that the installation of Microsoft Edge and Windows Admin Center completed.

9. Close the other two instances of the **Administrator: Windows PowerShell ISE** window you opened earlier in this task without saving the scripts you ran from each.

18.1.3 Task 3: Deploy Storage Spaces Direct clusters

1. In the console session to the **WSLab-Management** VM, from the **script** pane of the **Administrator: Windows PowerShell ISE** window, run the following command to provision Storage Spaces Direct clusters:

```
$clusters=@()
$clusters+=@{Nodes=1..2 | % {"2T2node$_"} ; Name="2T2nodeClus" ; IP="10.0.0.112" }
$clusters+=@{Nodes=1..3 | % {"2T3node$_"} ; Name="2T3nodeClus" ; IP="10.0.0.113" }
$clusters+=@{Nodes=1..2 | % {"3T2node$_"} ; Name="3T2nodeClus" ; IP="10.0.0.115" }
```

```
$clusters+=@{Nodes=1..3 | % {"3T3node$_"} ; Name="3T3nodeClus" ; IP="10.0.0.116" }

# Install features on servers
Invoke-Command -computername $clusters.nodes -ScriptBlock {
  Install-WindowsFeature -Name "Failover-Clustering","Hyper-V-PowerShell","RSAT-Clustering-PowerSh
}

# Restart servers since failover clustering in Windows Server 2019 requires reboot
Restart-Computer -ComputerName $clusters.nodes -Protocol WSMan -Wait -For PowerShell

# Create clusters
foreach ($cluster in $clusters){
  New-Cluster -Name $cluster.Name -Node $cluster.Nodes -StaticAddress $cluster.IP
  Start-Sleep 5
  Clear-DNSClientCache
}

# Add file share witness
foreach ($cluster in $clusters){
  $clusterName = $cluster.Name
  # Create new directory
  $WitnessName = $clusterName+"Witness"
  Invoke-Command -ComputerName DC -ScriptBlock {New-Item -Path c:\Shares -Name $using:WitnessName
  $accounts = @()
  $accounts += "CORP\$($clusterName)$"
  $accounts += "CORP\Domain Admins"
  New-SmbShare -Name $WitnessName -Path "c:\Shares\$WitnessName" -FullAccess $accounts -CimSession
  # Set NTFS permissions
  Invoke-Command -ComputerName DC -ScriptBlock {(Get-SmbShare $using:WitnessName).PresetPathAcl |
  # Set Quorum
  Set-ClusterQuorum -Cluster $clusterName -FileShareWitness "\\DC\$WitnessName"
}

# Enable Storage Spaces Direct and configure mediatype to simulate 3 tier system with SCM (all 800
foreach ($cluster in $clusters.Name){
  Enable-ClusterS2D -CimSession $cluster -Verbose -Confirm:0
  if ($cluster -like "3T*"){
    invoke-command -computername $cluster -scriptblock {
      Get-PhysicalDisk | Where-Object size -eq 800GB | Set-PhysicalDisk -MediaType SCM
      Get-PhysicalDisk | Where-Object size -eq 4TB | Set-PhysicalDisk -MediaType SSD
    }
  }
}
```

> **Note:** Wait for the script to complete before you proceed to the next task. The script should take about 10 minutes to complete. Disregard any errors or warnings.

2. To generate a text file, you will use to import the list of Storage Spaces Direct clusters into Windows Admin Center, in the console session to the **WSLab-Management** VM, from the **console** pane of the **Administrator: Windows PowerShell ISE** window, run the following command:

   ```
   (Get-Cluster -Domain $env:userdomain | Where-Object S2DEnabled -eq 1).Name | Out-File c:\s2dcluste
   ```

3. Switch to the Microsoft Edge browser window, select **Get started**, accept the default tab page settings, select the **Continue without Signing-in** link, use the Microsoft Edge browser to navigate to `https://management.corp.contoso.com`, and then when prompted to authenticate, sign in as **CORP\LabAdmin** with the password **LS1setup!**.

4. In the console session to the **WSLab-Management** VM, in the browser window displaying the Windows Admin Center interface, on the **All connections** page, select **+ Add**.

5. On the **Add or create resources** panel, on the **Server clusters** tile, select **Add**, select the **Import clusters** tab, select **Select a file**, in the **Open** dialog box, locate the **c:\s2dclusters.txt** file, select **Open**, and then select **Add**.

18.1.4 Task 4: Configure Storage Spaces Direct cluster tiers

1. In the console session to the **WSLab-Management** VM, from the **script** pane of the **Administrator: Windows PowerShell ISE** window, run the following command to identify the tiers on a two-node cluster with HDDs only:

```
Get-StorageTier -CimSession 2T2NodeClus |
  ft FriendlyName,MediaType,ResiliencySettingName,NumberOfDataCopies,PhysicalDiskRedundancy,FaultD
```

   Note: On a two-node Storage Spaces Direct cluster with HDDs only, there are two tiers. One is **Capacity**, created to provide compatibility with the naming convention in Windows Server 2016, and the other is **MirrorOnHDD**, which follows the naming convention in Windows Server 2019. The value of **NumberOfDataCopies** represents the **2way mirror** configuration and **PhysicalDiskRedundancy** reflects the ability to tolerate a single fault. The value of **FaultDomainAwareness** indicates that the two copies are distributed across instances of **StorageScaleUnit**. The number of columns is automatically calculated, depending on the number of nodes and disks in each node, and is assigned during the creation of the virtual disks. The value of **NumberOfGroups** indicates the parity setting, which in this case is set to **1**.

2. In the console session to the **WSLab-Management** VM, from the **script** pane of the **Administrator: Windows PowerShell ISE** window, run the following command to identify all of the tiers on all of the Storage Spaces Direct clusters in the lab environment:

```
$clusters=(Get-Cluster -Domain $env:userdomain | Where-Object S2DEnabled -eq 1).Name
Get-StorageTier -CimSession $clusters |
  Sort-Object PSComputerName |
  ft PSComputerName,FriendlyName,MediaType,ResiliencySettingName,NumberOfDataCopies,PhysicalDiskRe
```

   Note: The tiers are generated automatically when you invoke the **Enable-ClusterS2D** PowerShell cmdlet. Tiers reflect the media type present in cluster nodes.

3. In the console session to the **WSLab-Management** VM, from the **script** pane of the **Administrator: Windows PowerShell ISE** window, run the following command to identify the Windows Server 2019-specific mirror tiers on all of the Storage Spaces Direct clusters in the lab environment (tiers that reference **MirrorOnHDD**, **MirrorOnSSD**, and **MirrorOnSCM**, where *SCM* designates Storage Class Memory):

```
$clusters=(Get-Cluster -Domain $env:userdomain | Where-Object S2DEnabled -eq 1).Name
Get-StorageTier -CimSession $clusters |
  Where-Object friendlyname -like mirror* |
  Sort-Object PSComputerName |
  ft PSComputerName,FriendlyName,MediaType,ResiliencySettingName,NumberOfDataCopies,PhysicalDiskRe
```

   Note: The values of **NumberOfCopies** and **PhysicalDiskRedundancy** is **2** on two-node clusters and **3** for clusters with three or more nodes.

4. In the console session to the **WSLab-Management** VM, from the **script** pane of the **Administrator: Windows PowerShell ISE** window, run the following command to identify the Windows Server 2019-specific parity tiers on all of the Storage Spaces Direct clusters in the lab environment (tiers that reference **MirrorOnHDD**, **MirrorOnSSD**, and **MirrorOnSCM**, where *SCM* designates Storage Class Memory):

```
$clusters=(Get-Cluster -Domain $env:userdomain | Where-Object S2DEnabled -eq 1).Name
Get-StorageTier -CimSession $clusters |
  Where-Object friendlyname -like parity* |
  Sort-Object PSComputerName |
  ft PSComputerName,FriendlyName,MediaType,ResiliencySettingName,NumberOfDataCopies,PhysicalDiskRe
```

   Note: The script does not return any results, because by default, parity tiers are created only on Storage Spaces Direct clusters with four or more nodes. With Windows Server 2019-based Storage Spaces Direct clusters, you have the option to create parity tiers by implementing nested resiliency on two-node clusters.

5. In the console session to the **WSLab-Management** VM, from the **script** pane of the **Administrator: Windows PowerShell ISE** window, run the following command to create nested resiliency tiers:

```
#Select clusters to fix tiers
$clusterNames=(Get-Cluster -Domain $env:userdomain | Where-Object S2DEnabled -eq 1 | Out-GridView -

foreach ($clusterName in $clusterNames){
   $storageTiers=Get-StorageTier -CimSession $clusterName
   $numberOfNodes=(Get-ClusterNode -Cluster $clusterName).Count
   $MediaTypes=(Get-PhysicalDisk -CimSession $clusterName |where mediatype -ne Unspecified | Where-
   $clusterFunctionalLevel=(Get-Cluster -Name $clusterName).ClusterFunctionalLevel

   foreach ($mediaType in $mediaTypes){
      if ($numberOfNodes -eq 2) {
         # Create Mirror Tiers
         if (-not ($storageTiers | Where-Object FriendlyName -eq "MirrorOn$mediaType")){
            New-StorageTier -CimSession $clusterName -StoragePoolFriendlyName "S2D on $clusterName
         }
         if ($clusterFunctionalLevel -ge 10){
            # Create NestedMirror Tiers
            if (-not ($storageTiers | Where-Object FriendlyName -eq "NestedMirrorOn$mediaType")){
               New-StorageTier -CimSession $clusterName -StoragePoolFriendlyName "S2D on $clusterNa
            }
            #Create NestedParity Tiers
            if (-not ($storageTiers | Where-Object FriendlyName -eq "NestedParityOn$mediaType")){
               New-StorageTier -CimSession $clusterName -StoragePoolFriendlyName "S2D on $clusterNa
            }
         }
      }
   }
}
```

> **Note:** When prompted, in the **Select Clusters to Check on tiers** window, select the **Ctrl** key, select both the **2T2nodeClus** and **3T2nodeClus** entry, and then select **OK**.

6. In the console session to the **WSLab-Management** VM, from the **script** pane of the **Administrator: Windows PowerShell ISE** window, run the following command to identify the Windows Server 2019-specific nested tiers on two-node Storage Spaces Direct clusters in the lab environment:

```
$clusters=(Get-Cluster -Domain $env:userdomain | Where-Object S2DEnabled -eq 1).Name
Get-StorageTier -CimSession $clusters |
  Where-Object friendlyname -like nested* |
  Sort-Object PSComputerName |
  ft PSComputerName,FriendlyName,MediaType,ResiliencySettingName,NumberOfDataCopies,PhysicalDiskRe
```

> **Note:** By default, parity tiers are created only on Storage Spaces Direct clusters with four or more nodes. With Windows Server 2019-based Storage Spaces Direct clusters, you have the option to create parity tiers by implementing nested resiliency on two-node clusters.

18.1.5 Task 5: Provision nested tier volumes

1. In the console session to the **WSLab-Management** VM, from the **script** pane of the **Administrator: Windows PowerShell ISE** window, run the following command to provision a volume on the Storage Spaces Direct cluster **2T2nodeClus** based on the **NestedMirrorOnHDD** nested resiliency tier:

```
$clusterName = '2T2nodeClus'
New-Volume -StoragePoolFriendlyName s2d* -FriendlyName NestedMirroronHDDVolume -FileSystem CSVFS_Re
```

> **Note:** Wait until the volume is provisioned. This should take less than one minute.

2. In the console session to the **WSLab-Management** VM, in the browser window displaying the Windows Admin Center interface, on the **All connections** page, select **2T2nodeClus**.

3. On the **Specify your credentials** panel, select the **Use another account for this connection** option, and then in the **Username** text box, enter **CORP\LabAdmin**, and in the **Password** text box, enter **LS1setup!**, select the **Use these credentials for all connections**, and then select **Continue**.

4. On the **2T2nodeClus** page, in the **Storage** section, select **Volumes**, and then on the **Volumes** pane, select the **Inventory** tab.

5. In the list of volumes, note that the **NestedMirroronHDDVolume** has resiliency set to **Nested two-way mirror**.

6. In the browser window displaying the Windows Admin Center interface, navigate back to the **2T2nodeClus** page.
7. In the **Storage** section, select **Drives**, and then on the **Drives** pane, select the **Inventory** tab.
8. In the list of drives, review the list of drive types.
9. In the console session to the **WSLab-Management** VM, from the **script** pane of the **Administrator: Windows PowerShell ISE** window, run the following command to provision a volume on the Storage Spaces Direct cluster **3T2nodeClus** based on the **NestedMirrorOnSSD** nested resiliency tier:

   ```
   $clusterName = '3T2nodeClus'
   New-Volume -StoragePoolFriendlyName s2d* -FriendlyName NestedMirroronSSDVolume -FileSystem CSVFS_R
   ```

10. In the console session to the **WSLab-Management** VM, in the browser window displaying the Windows Admin Center interface, navigate back to the **All connections** page, and then select **3T2nodeClus**.
11. On the **3T2nodeClus** page, in the **Storage** section, select **Volumes**, and then on the **Volumes** pane, select the **Inventory** tab.
12. In the list of volumes, note that the **NestedMirroronSSDVolume** has the resiliency set to **Nested two-way mirror**.
13. In the browser window displaying the Windows Admin Center interface, navigate back to the **3T2nodeClus** page, in the **Storage** section, select **Drives**, and then on the **Drives** pane, select the **Inventory** tab.
14. In the list of drives, review the list of drive types.

18.1.6 Task 6: Deprovision the lab resources

1. Switch to the lab VM, and in the **Administrator: Windows PowerShell ISE** window, open and run the **F:\WSLab-master\Scripts\Cleanup.ps1** script to remove all VMs provisioned in this lab. When prompted, on the **console** pane, enter **Y**, and then select **Enter**.
2. When the script completes, select any key.
3. In the **Administrator: Windows PowerShell ISE** window, close the tab displaying the **F:\WSLab-master\Scripts\Cleanup.ps1** script. --- lab: title: 'Lab E: Identifying and analyzing metadata of a Storage Spaces Direct cluster (optional)' type: 'Answer Key' module: 'Module 3: Planning for and implementing Azure Stack HCI Storage'

---

19 Lab E answer key: Identifying and analyzing metadata of a Storage Spaces Direct cluster (optional)

19.1 Exercise 1: Identifying and analyzing metadata of a Storage Spaces Direct cluster

19.1.1 Task 1: Provision the lab environment VMs

1. From the lab VM, in the **script** pane of the **Administrator: Windows PowerShell ISE** window, run the following command to rename **LabConfig.ps1** and **Scenario.ps1**:

   ```
   Set-Location -Path 'F:\WSLab-master\Scripts'
   Move-Item -Path '.\LabConfig.ps1' -Destination '.\LabConfig.m315.ps1' -Force -ErrorAction Silently
   Move-Item -Path '.\Scenario.ps1' -Destination '.\Scenario.m315.ps1' -Force -ErrorAction SilentlyCo
   ```

2. In the **Administrator: Windows PowerShell ISE** window, open a new tab, and in the **script** pane, paste the following command, and then save it as **F:\WSLab-master\Scripts\LabConfig.ps1**:

   ```
   $LabConfig=@{ DomainAdminName = 'LabAdmin'; AdminPassword = 'LS1setup!'; Prefix = 'WSLab-'; Switch
   1..6 | % {$VMNames = "6node"; $LABConfig.VMs += @{ VMName = "$VMNames$_" ; Configuration = 'S2D' ;
   $LABConfig.VMs += @{
        VMName = "Management" ;
        Configuration = 'S2D' ;
        ParentVHD = 'Win2019_G2.vhdx';
        SSDNumber = 1;
   ```

```
    SSDSize = 50GB ;
    MemoryStartupBytes = 8GB;
    NestedVirt = $false;
    StaticMemory = $true;
    VMProcessorCount = 4
}
```

3. On the lab VM, in the **Administrator: Windows PowerShell ISE** window, open and run the **F:\WSLab-master\Scripts\3_Deploy.ps1** script to provision VMs for the Storage Spaces Direct environment.

   > **Note:** Select **None** for the Telemetry prompt. The script should complete in about 15 minutes. When prompted **Press enter to continue**, select the **Enter** key.

4. When the script completes, in the **Administrator: Windows PowerShell ISE** window, from the **console** pane, run the following command to start the newly provisioned VMs that will host the Storage Spaces Direct environment:

   ```
   Get-VM -Name 'WSLab-Management' | Start-VM
   Start-Sleep 150
   Get-VM | Where-Object Name -like 'WSLab-*node*' | Start-VM -AsJob
   ```

5. On the lab VM, start **Hyper-V Manager** and connect via a console session to **WSLab-DC**. When prompted to sign in, provide the username **CORP\LabAdmin** and the password **LS1setup!**.

6. In the **WSLab-DC** VM console session, start **Windows PowerShell ISE** as an administrator.

7. From the **Administrator: Windows PowerShell ISE** window, run `slmgr -rearm` and then select **OK**.

8. From the **Administrator: Windows PowerShell ISE** window, run `Restart-Computer -Force`.

   Note: Make sure that the **WSLab-DC** VM is running before you proceed to the next task.

19.1.2 Task 2: Deploy a Storage Spaces Direct cluster

1. On the lab VM, in the **Server Manager** window, select **Tools**, and then in the drop-down list, select **Hyper-V Manager**.

2. On the lab VM, in the **Hyper-V Manager** console, in the list of virtual machines, right-click or access the context menu for the **WSLab-Management** entry, and then select **Connect** to establish a console session to the **WSLab-Management** VM. When prompted to sign in, provide the username **CORP\LabAdmin** and the password **LS1setup!**.

3. In the console session to the **WSLab-Management** VM, start Windows PowerShell ISE as Administrator.

4. In the console session to the **WSLab-Management** VM, from the **script** pane of the **Administrator: Windows PowerShell ISE** window, run the following command to provision a six-node Storage Spaces Direct cluster:

   ```
   $clusters = @()
   $clusters += @{Nodes=1..6 | % {"6node$_"} ; Name="6nodeCluster" ; IP="10.0.0.116" }

   Install-WindowsFeature -Name RSAT-Clustering,RSAT-Clustering-Mgmt,RSAT-Clustering-PowerShell,RSAT-

   # Install features on servers
   Invoke-Command -computername $clusters.nodes -ScriptBlock {
      Install-WindowsFeature -Name "Failover-Clustering","Hyper-V-PowerShell"
   }

   # Restart all servers to finalize installation of Failover Clustering
   Restart-Computer -ComputerName $clusters.nodes -Protocol WSMan -Wait -For PowerShell

   # Create clusters
   foreach ($cluster in $clusters){
      New-Cluster -Name $cluster.Name -Node $cluster.Nodes -StaticAddress $cluster.IP
      Start-Sleep 5
   ```

```
   Clear-DNSClientCache
}

# Add file share witness
foreach ($cluster in $clusters){
   $clusterName = $cluster.Name
   # Create new directory
   $witnessName = $clusterName+"Witness"
   Invoke-Command -ComputerName DC -ScriptBlock {New-Item -Path c:\Shares -Name $using:witnessName
   $accounts = @()
   $accounts += "CORP\$($clusterName)$"
   $accounts += "CORP\Domain Admins"
   New-SmbShare -Name $witnessName -Path "c:\Shares\$witnessName" -FullAccess $accounts -CimSessio
   # Set NTFS permissions
   Invoke-Command -ComputerName DC -ScriptBlock {(Get-SmbShare $using:witnessName).PresetPathAcl |
   # Set Quorum
   Set-ClusterQuorum -Cluster $clusterName -FileShareWitness "\\DC\$WitnessName"
}

# Enable S2D
Enable-ClusterS2D -CimSession $clusters.Name -Verbose -Confirm:0
```

Note: Wait for the script to complete. This should take about five minutes.

19.1.3 Task 3: Examine physical disk owners

1. In the console session to the **WSLab-Management** VM, from the **console** pane of the **Administrator: Windows PowerShell ISE** window, run the following command to display the physical disks of the **6nodecluster** Storage Spaces Direct cluster:

```
Get-PhysicalDisk -CimSession 6nodecluster |ft FriendlyName,Size,Description
```

2. In the console session to the **WSLab-Management** VM, from the **script** pane of the **Administrator: Windows PowerShell ISE** window, run the following command to set the **Description** attribute of each physical disk to the name of the cluster node to which the disk is attached for the Storage Spaces Direct cluster:

```
$clusters = @()
$clusters += @{Nodes=1..6 | % {"6node$_"} ; Name="6nodeCluster" ; IP="10.0.0.116" }
foreach ($clusterName in ($clusters.Name | select -Unique)){
   $storageNodes=Get-StorageSubSystem -CimSession $clusterName -FriendlyName Clus* | Get-StorageNo
   foreach ($storageNode in $storageNodes){$storageNode | Get-PhysicalDisk -PhysicallyConnected -C
}
```

3. In the console session to the **WSLab-Management** VM, from the **console** pane of the **Administrator: Windows PowerShell ISE** window, rerun the following command to display physical disks of **6nodecluster** Storage Spaces Direct cluster, this time with the **Description** attribute containing the name of the owner node:

```
Get-PhysicalDisk -CimSession 6nodecluster |ft FriendlyName,Size,Description
```

19.1.4 Task 4: Explore metadata of the storage pool

1. In the console session to the **WSLab-Management** VM, from the **console** pane of the **Administrator: Windows PowerShell ISE** window, run the following command to display the number of disks with metadata for the six-node cluster:

```
foreach ($clusterName in ($clusters.Name | select -Unique)){
  Get-StoragePool -CimSession $clusterName |
  Get-PhysicalDisk -HasMetadata -CimSession $clusterName |
  Sort-Object Description |
  Format-Table DeviceId,FriendlyName,SerialNumber,MediaType,Description
}
```

Note: The number of disks with metadata depends on the size of the cluster, as the following table displays:

Table 1: The number of disks with metadata

Number of nodes (fault domains)	Number of disks with metadata
2	6
3	6
4	8
5	5
6	5

Note: In a six-node cluster, metadata is located on five of the six nodes. This means that if you lose random half nodes, 50 percent of the storage pool will go offline.

2. In the console session to the **WSLab-Management** VM, switch to the **Server Manager** window, select **Tools**, and then in the **Tools** drop-down list, select **Failover Cluster Manager**.
3. In the **Failover Cluster Manager** window, right-click or access the context menu for the **Failover Cluster Manager** node, and then in the context menu, select **Connect to cluster**.
4. In the **Select Cluster** dialog box, in the **Cluster name** text box, enter `6nodecluster.corp.contoso.com`, and then select **OK**.
5. In the **Failover Cluster Manager** window, select **Nodes**, and then review the list of nodes.
6. In the **Failover Cluster Manager** window, in the **Storage** node tree, select **Pools**, and then verify that it contains a single pool named **Cluster Pool 1**.
7. Select the **Cluster Pool 1** entry, and on the **Cluster Pool 1** pane, examine its properties by selecting the **Summary** tab, followed by the **Virtual Disks** and **Physical Disks** tabs.
8. In the console session to the **WSLab-Management** VM, switch to the **Administrator: Windows PowerShell ISE** window, and then from the **script** pane, run the following command to capture the list of three nodes hosting the storage pool metadata of the six-node cluster:

```
$nodesToShutDown = (Get-StoragePool -CimSession 6nodecluster |
Get-PhysicalDisk -HasMetadata -CimSession $clusterName | Select-Object -First 3).Description
```

9. In the console session to the **WSLab-Management** VM, from the **console** pane of the **Administrator: Windows PowerShell ISE** window, run the following command to shut down the three nodes hosting the metadata of the six-node cluster:

```
Stop-Computer -ComputerName $nodesToShutDown -Force
```

10. In the console session to the **WSLab-Management** VM, switch to the **Failover Cluster Manager** window, and then in the **Storage** node tree, with the **Pools** node selected, verify that the **Cluster Pool 1** storage pool has a status of **Failed**.

    Note: It might take a few minutes before the storage pool reaches the **Failed** status.

11. In the **Failover Cluster Manager** window, on the **Actions** pane, select **Show Critical Events**, and in the list of events, locate the most recent event that references the storage pool failure because of the lack of quorum of healthy disks.
12. In the **Critical events for Cluster Pool 1** window, review the event, and then select **Close**.
13. Switch to the lab VM, and then in the **Hyper-V Manager** console, in the list of virtual machines, select the VMs you shut down earlier in this task, and then in the **Actions** pane, in the **Selected Virtual Machines** section, select **Start**.
14. Switch to the **WSLab-Management** VM, in the **Failover Cluster Manager** window, right-click or access the context menu for the **Cluster Pool 1** entry, and in the context menu, select **Bring Online**.

19.1.5 Task 5: Explore metadata of a volume

1. In the console session to the **WSLab-Management** VM, from the **console** pane of the **Administrator: Windows PowerShell ISE** window, run the following command to create a volume on the six-node cluster:

```
Invoke-Command -ComputerName ($clusters.Name | select -Unique) -ScriptBlock {New-Volume -FriendlyNa
```

2. In the console session to the **WSLab-Management** VM, from the **console** pane of the **Administrator: Windows PowerShell ISE** window, run the following command to display the metadata of the newly created volume on the six-node cluster:

```
foreach ($clusterName in ($clusters.Name | select -Unique)){
   Get-VirtualDisk -FriendlyName labVolume -CimSession $clusterName |
   Get-PhysicalDisk -HasMetadata -CimSession $clusterName |
   Sort-Object Description |
   Format-table DeviceId,FriendlyName,SerialNumber,MediaType,Description
}
```

> **Note:** Based on the output, you can determine whether the number of disks with metadata matches the number of disks used by the storage pool metadata.

19.1.6 Task 6: Explore metadata of a scoped volume

1. In the console session to the **WSLab-Management** VM, from the **script** pane of the **Administrator: Windows PowerShell ISE** window, run the following command to create scoped volumes on the six-node cluster:

```
$faultDomains = Get-StorageFaultDomain -Type StorageScaleUnit -CimSession 6nodecluster | Sort Frie
New-Volume -FriendlyName "2Scope-Volume" -Size 100GB -StorageFaultDomainsToUse ($faultDomains | Ge
New-Volume -FriendlyName "3Scope-Volume" -Size 100GB -StorageFaultDomainsToUse ($faultDomains | Ge
New-Volume -FriendlyName "4Scope-Volume" -Size 100GB -StorageFaultDomainsToUse ($faultDomains | Ge
New-Volume -FriendlyName "5Scope-Volume" -Size 100GB -StorageFaultDomainsToUse ($faultDomains | Ge
New-Volume -FriendlyName "6Scope-Volume" -Size 100GB -StorageFaultDomainsToUse ($faultDomains | Ge
```

> **Note:** For a six-node cluster, set the number of a volume's scopes to four. The additional volumes in this exercise aren't used as an example of their practical use, but rather as an illustration about how different scope values affect volume distribution.

2. In the console session to the **WSLab-Management** VM, from the **script** pane of the **Administrator: Windows PowerShell ISE** window, run the following command to display the metadata of the newly created volumes on the six-node cluster:

```
$friendlyNames=2..6 | % {"$($_)Scope-Volume"}
foreach ($friendlyName in $friendlyNames){
   Write-Host -Object "$friendlyName" -ForeGroundColor Cyan
   Get-VirtualDisk -FriendlyName $friendlyName -CimSession 6nodecluster |
   Get-PhysicalDisk -HasMetadata -CimSession 6nodecluster |
   Sort-Object Description |
   Format-Table DeviceId,FriendlyName,SerialNumber,MediaType,Description
}
```

> **Note:** Review the output and note that the number of cluster nodes containing metadata of each volume matches the scope determined by the value of the **StorageFaultDomainsToUse** parameter of the **New-Volume** cmdlet.

19.1.7 Task 7: Deprovision the lab resources

1. Switch to the lab VM, and in the **Administrator: Windows PowerShell ISE** window, open and run the **F:\WSLab-master\Scripts\Cleanup.ps1** script to remove all VMs provisioned in this lab. When prompted, on the **console** pane, enter **Y**, and then select **Enter**.
2. After the script completes, select any key.
3. In the **Administrator: Windows PowerShell ISE** window, close the tab displaying the **F:\WSLab-master\Scripts\Cleanup.ps1** script.

19.2 lab: title: 'Lab A: Deploying Software-Defined Networking' type: 'Answer Key' module: 'Module 4: Planning for and Implementing Azure Stack HCI Networking'

20 Lab A answer key: Deploying Software-Defined Networking

20.1 Exercise 1: Deploying Software-Defined Networking by using PowerShell

20.1.1 Task 1: Deploy the VMs that will serve as the SDN infrastructure Hyper-V hosts

1. On the lab virtual machine (VM), start Windows PowerShell Integrated Scripting Environment (ISE) as Administrator, and from the **console** pane, run the following to remove the **Zone.Identifier** alternate data stream, which has a value of **3** indicating that it was downloaded from the internet:

   ```
   Get-ChildItem -Path F:\WSLab-master\ -File -Recurse | Unblock-File
   ```

2. On the lab VM, from the **console** pane of the Administrator: Windows PowerShell ISE window, run the following to set the current directory:

   ```
   Set-Location -Path F:\WSLab-master\Scripts
   ```

3. On the lab VM, from the **script** pane of the Administrator: Windows PowerShell ISE window, run the following commands to rename **LabConfig.ps1** and **Scenario.ps1**.

   ```
   Move-Item -Path '.\LabConfig.ps1' -Destination '.\LabConfig.m410.ps1' -Force -ErrorAction SilentlyC
   Move-Item -Path '.\Scenario.ps1' -Destination '.\Scenario.m410.ps1' -Force -ErrorAction SilentlyCo
   ```

4. On the lab VM, in the Administrator: Windows PowerShell ISE window, open a new tab in the **script** pane, paste the following content, and then save it as **F:\WSLab-master\Scripts\LabConfig.ps1**:

   ```
   $LabConfig=@{ DomainAdminName = 'LabAdmin'; AdminPassword = 'LS1setup!'; Prefix = 'SDNExpress2019-
   $LABConfig.AdditionalNetworksConfig += @{
        NetName = 'HNV';
        NetAddress = '10.103.33.';
        NetVLAN = '201';
        Subnet = '255.255.255.0'
    }

   1..4 | % {
   $VMNames = "HV";
   $LABConfig.VMs += @{
        VMName = "$VMNames$_";
        Configuration = 'S2D';
        ParentVHD = 'Win2019Core_G2.vhdx';
        SSDNumber = 2;
        SSDSize = 800GB;
        HDDNumber = 4;
        HDDSize = 4TB;
        MemoryStartupBytes = 20GB;
        NestedVirt = $True;
        StaticMemory = $True;
        VMProcessorCount = 6
       }
   }

   $LABConfig.VMs += @{
        VMName = "Management";
        Configuration = 'S2D';
        ParentVHD = 'Win2019_G2.vhdx';
        SSDNumber = 1;
        SSDSize = 50GB;
        MemoryStartupBytes = 4GB;
        NestedVirt = $false;
        StaticMemory = $false;
   ```

```
    VMProcessorCount = 4
}
```

5. On the lab VM, in the Administrator: Windows PowerShell ISE window, from the **script** pane, run the following commands to copy the **Scenario.ps1** and **MultiNodeConfig.psd1** files from **F:\WSLab-master\Scenarios\SDNExpress with Windows Admin Center** to the current directory:

```
Copy-Item -Path 'F:\WSLab-master\Scenarios\SDNExpress with Windows Admin Center\Scenario.ps1' -Des
Copy-Item -Path 'F:\WSLab-master\Scenarios\SDNExpress with Windows Admin Center\MultiNodeConfig.ps
```

6. In the Administrator: Windows PowerShell ISE window open and run the **F:\WSLab-master\Scripts\3_Deploy.ps** script to provision VMs for the SDN environment.

   > **Note**: The script should complete in about 7 minutes. When prompted **Press enter to continue**, select the **Enter** key.

7. In the Administrator: Windows PowerShell ISE window, open the **F:\WSLab-master\Scripts\Scenario.ps1** script, remove all content following the line **128**, starting from `# ENDING Run from Hyper-V Host ENDING #`, and then save the modified file as **Scenario_Part1.ps1**.

   > **Note**: This part of the scenario script needs to be run from the Hyper-V host.

8. From the Administrator: Windows PowerShell ISE window, run the newly saved **F:\WSLab-master\Scripts\Scenario_Part1.ps1** script to configure the VMs that will host the lab environment. When prompted for the location of the parent virtual hard disk (VHDX) for SDN VMs, point to **F:\WSLab-master\Scripts\ParentDisks\Win2019Core_G2.vhdx**. When prompted for the **MultiNodeConfig.psd1** file, point to the file you copied to **F:\WSLab-master\Scripts**. When prompted for Windows Admin Center MSI, point to the downloaded Windows Installer file in the **F:\Source** folder.

   > **Note**: If the script fails with the message **Copy-VMFile : Failed to initiate copying files to the guest**, rerun the script.

   > **Note**: The script should complete in about 15 minutes.

   > **Note**: Ignore the error following the line **ScriptHalted** and message prompting to restart **SDNExpress2019-Management**. That's expected.

9. After the script completes, in the Administrator: Windows PowerShell ISE window, open a new tab, and run the following script to expand the size of the disks hosting drive **C** of the newly provisioned VMs that will host the SDN environment:

```
$servers = @('SDNExpress2019-HV1','SDNExpress2019-HV2','SDNExpress2019-HV3','SDNExpress2019-HV4')
$paths = (Get-VM -Name $servers | Get-VMHardDiskDrive | Where-Object {$_.ControllerLocation -eq 0}
foreach ($path in $paths) { Resize-VHD -Path $path -SizeBytes 100GB }
```

20.1.2 Task 2: Deploy the SDN infrastructure VMs

> **Note**: Sign in to the **DC** VM using the **CORP\LabAdmin** username and **LS1setup!** password, run `slmgr -rearm` and restart it.

1. On the lab VM, start the **Hyper-V Manager** console and establish a console session to the **SDNExpress2019-Management** VM. When prompted to sign in, provide the **CORP\LabAdmin** username and **LS1setup!** password.

2. Within the console session to the **SDNExpress2019-Management** VM, start Windows PowerShell ISE as Administrator.

3. Within the console session to the **SDNExpress2019-Management** VM, in the Administrator: Windows PowerShell ISE window, in the **script** pane, open a new tab, paste the following script, and then run it to expand the size of drive **C** of the VMs that will host the SDN environment:

```
$servers = @('HV1','HV2','HV3','HV4')
Invoke-Command -ComputerName $servers -ScriptBlock {
  $size = Get-PartitionSupportedSize -DriveLetter C
  Resize-Partition -DriveLetter C -Size $size.SizeMax
}
```

4. On the lab VM, from the **script** pane of the Administrator: Windows PowerShell ISE window, run the following commands to download the following file:

```
New-Item F:\Allfiles -itemtype directory -Force
Invoke-Webrequest -Uri "https://raw.githubusercontent.com/MicrosoftLearning/WS-013T00-Azure-Stack-
```

5. Within the console session to the **SDNExpress2019-Management** VM, start File Explorer and navigate to the **C:\Library** folder.
6. Switch back to the lab VM and use the copy and paste functionality of the **Hyper-V** console session to copy **F:\WSLab-master\Scripts\Scenario.ps1** and **F:\Allfiles\SDNExpressModule.psm1** on the lab VM to **C:\Library** on the **SDNExpress2019-Management VM**.
7. Within the console session to the **SDNExpress2019-Management** VM, in the Administrator: Windows PowerShell ISE window, open the **C:\Library\Scenario.ps1** script, and comment out line 375 so it looks like so: `# Expand-Archive -Path C:\SDN-Master.zip -DestinationPath C:\Library`
8. Within the console session to the **SDNExpress2019-Management** VM, in the Administrator: Windows PowerShell ISE window, remove all content before the line 136, up to the line prior to `# Run from DC / VMM #`, and then save the modified file as **Scenario_Part2.ps1**.
9. Within the console session to the **SDNExpress2019-Management** VM, in the Administrator: Windows PowerShell ISE window, run the following command:

```
Expand-Archive -Path C:\SDN-Master.zip -DestinationPath C:\Library
Copy-Item -Path C:\Library\SDNExpressModule.psm1 -Destination C:\Library\SDN-master\SDNExpress\scr
```

> **Note**: This part of the scenario script needs to be run from the management VM.

10. Within the console session to the **SDNExpress2019-Management** VM, in the Administrator: Windows PowerShell ISE window, run the newly saved **C:\Library\Scenario_Part2.ps1** script to configure the SDN VMs.

> **Note**: Wait until the script completes before you proceed. The script should complete in about 90 minutes. Disregard any cluster validation errors.

20.2 lab: title: 'Lab B: Managing virtual networks by using Windows Admin Center and PowerShell' type: 'Answer Key' module: 'Module 4: Planning for and Implementing Azure Stack HCI Networking'

21 Lab B answer key: Managing virtual networks by using Windows Admin Center and PowerShell

21.1 Exercise 1: Managing virtual networks by using Windows Admin Center and PowerShell

21.1.1 Task 1: Connect to the SDN infrastructure by using Windows Admin Center

1. Within the console session to the **SDNExpress2019-Management** virtual machine (VM), switch to the File Explorer window displaying the content of the **C:\Library** folder and use the **chrome_installer.exe** to install the Chrome browser.
2. In the Chrome browser, navigate to `https://management:9999`. If prompted to authenticate, sign in as **CORP\LabAdmin** with **LS1setup!** as the password.

> **Note**: This URL designates the local installation of **Windows Admin Center** on the management VM.

3. In the **Windows Admin Center** interface, on the **All connections** page, select **+ Add**. On the **Add resources** panel, select **Add** in the **Windows Server cluster** tile, and in the **Cluster name** text box, enter `sddc01.corp.contoso.com`. If prompted, select the **Use another account for this connection**, in the **Username** text box, enter **CORP\LabAdmin**, and in the **Password** text box, enter **LS1setup!**. Select **Manage Software-Defined Networking (if set up)**. In the **Specify the Network Controller REST URI** box, enter `https://NCCLUSTER.corp.contoso.com`. If prompted, select **Connect with**

account, select **Validate**, select **Install-RSAT-NetworkController**, select **Validate** again, and then select **Add**.

> **Note**: If you are presented with the **Access was denied** message, select the option **Use another account for this connection**, and then select **OK**.

21.1.2 Task 2: Create virtual networks by using Windows Admin Center

1. In the console session to the **SDNExpress2019-Management** VM, in the **Windows Admin Center** interface, on the **All connections** page, select `sddc01.corp.contoso.com`.
2. On the `sddc01.corp.contoso.com` page, in the list of **Tools**, in the **Networking** section, select **Virtual switches** and review virtual switches on the members of the Software-Defined Network (SDN) cluster `sddc01.corp.contoso.com`.
3. Select the first **sdnSwitch** on `hv1.corp.contoso.com`, select **More**, and, in the drop-down list, select **Settings**.
4. Review general settings of **sdnSwitch** and note that you have the option of changing the **Load balancing algorithm** from **Hyper-V port** to **Dynamic**.
5. Select **Close** without making any changes.
6. On the `sddc01.corp.contoso.com` page, in the list of **Tools**, in the **Networking** section, select **Virtual networks** and, on the **Virtual networks** panel, select **Inventory**.
7. On the **Inventory** tab, select **+ New** and, on the **Virtual network** panel, specify the following settings:

   Table 1: vnet-000 settings

Setting	Value
Name	vnet-000
Address Prefix	192.168.0.0/20

8. On the **Virtual network** panel, in the **Subnets** section, select **+ Add** and, on the **Subnets** panel, specify the following settings:

   Table 2: vnet-000 subnet-0 settings

Setting	Value
Name	subnet-0
Address Prefix	192.168.0.0/24

9. Select **Submit**. Switch to the **Virtual network** panel, and in the **Subnets** section, select **+ Add** again.
10. On the **Subnets** panel, specify the following settings, and then select **Submit**:

    Table 3: vnet-000 subnet-1 settings

Setting	Value
Name	subnet-1
Address Prefix	192.168.1.0/24

11. On the **Virtual network** panel, select **Submit**.
12. Verify that **vnet-000** with two subnets was created successfully.
13. Back on the **Virtual networks** panel, select **+ New**.
14. Create another virtual network with the following settings:

    Table 4: vnet-100 settings

Setting	Value
Name	vnet-100
Address Prefix	192.168.96.0/20

15. Add to the virtual network **vnet-100** a single subnet with the following settings:

    Table 5: vnet-100 subnet-0 settings

Setting	Value
Name	subnet-0
Address Prefix	192.168.100.0/24

16. On the **Virtual network** panel, select **Submit**.
17. Verify that **vnet-100** with one subnet was created successfully.

21.1.3 Task 3: Create a storage volume on the hyperconverged cluster by using Windows Admin Center

1. Switch back to the lab VM and use the copy and paste functionality of the **Hyper-V** console session to copy the **ISO** image from the **F:\Source** folder to the **C:\Library** folder on the **SDNExpress2019-Management** VM.
2. In the browser window displaying the Windows Admin Center interface, on the `sddc01.corp.contoso.com` page, in the list of **Tools**, in the **Storage** section, select **Volumes**, and then select the **Inventory** tab.
3. On the **Inventory** tab of the **Volumes** panel, select **+ Create**.
4. On the **Create volume** panel, specify the following settings, and then select **Create**:

   Table 6: VMStorage volume settings

Setting	Value
Name	VMStorage
Resiliency	Mirror-accelerated parity
Parity percentage	90% parity, 10% mirror
Size on hard disk drive (HDD)	512
Size units	GB

5. On the **Create volume** panel, select the **Refresh** icon and ensure that the new volume appears in the **Inventory** listing.
6. Select the **VMStorage** entry, and on the **Volumes > Volume VMStorage** panel, select **Open**.

   > **Note**: This will automatically redirect you to the **Files** panel, which displays the content of the **VMStorage** volume.

7. On the **Files** panel, ensure that the **VMStorage** folder is selected, and in the toolbar, select **Upload**.
8. On the **Upload** panel, select **Select files**, in the **Open** dialog box, navigate to the **C:\Library** folder, select the ISO file, and select **Open**.
9. Back on the **Upload** panel, select **Submit**.

   > **Note**: Wait for the upload to complete. If the ISO file doesn't upload correctly, then from **SDNExpress2019-Management** virtual machine (VM), connect to **\\HV3\c$** and paste the ISO file into the **\\HV3\c$\ClusterStorage\VMStorage** folder.

21.1.4 Task 4: Create virtual machines by using Windows Admin Center

1. In the browser window displaying the Windows Admin Center interface, navigate back to the `sddc01.corp.contoso.com` page. In the **Tools** list, in the **Compute** section, select **Virtual machines**, and then select the **Inventory** tab.

2. On the **Inventory** tab, select + **New**.
3. On the **New virtual machine** panel, retain the default values for all other settings, specify the following settings, and then select **Create**:

   Table 7: vm-000 settings

Setting	Value
Name	vm-000
Generation	Generation 2 (Recommended)
Host	`hv3.corp.contoso.com`
Path	C:\ClusterStorage\VMStorage
Virtual processor count	2
Enable nested virtualization	Disabled
Startup memory (GB)	2
Network adapter	sdnSwitch
Connect to virtual network	Enabled
Virtual network	vnet-000
Virtual subnet	subnet-0 [192.168.0.0/24]
IP Address	192.168.0.100
Storage	Create an empty virtual hard disk
Size (GB)	64
Operating system	Install an operating system from an image file (.iso)
Path	Path to the ISO file you copied to the C:\ClusterStorage\VMStorage volume in the previous

4. Repeat the previous step to create an additional virtual machine with the following settings:

   Table 8: vm-001 settings

Setting	Value
Name	vm-001
Generation	Generation 2 (Recommended)
Host	`hv3.corp.contoso.com`
Path	C:\ClusterStorage\VMStorage
Virtual processor count	2
Enable nested virtualization	Disabled
Startup memory (GB)	2
Network adapter	sdnSwitch
Connect to virtual network	Enabled
Virtual network	vnet-000
Virtual subnet	subnet-1 [192.168.1.0/24]
IP Address	192.168.1.100
Storage	Create an empty virtual hard disk
Size (GB)	64
Operating system	Install an operating system from an image file (.iso)
Path	Path to the ISO file you copied to the C:\ClusterStorage\VMStorage volume in the previous

5. Repeat the previous step to create an additional virtual machine with the following settings:

   Table 9: vm-100 settings

Setting	Value
Name	vm-100
Generation	Generation 2 (Recommended)
Host	`hv3.corp.contoso.com`
Path	C:\ClusterStorage\VMStorage
Virtual processor count	2
Enable nested virtualization	Disabled
Startup memory (GB)	2

Setting	Value
Network adapter	sdnSwitch
Connect to virtual network	Enabled
Virtual network	vnet-100
Virtual subnet	subnet-0 [192.168.100.0/24]
IP Address	192.168.100.100
Storage	Create an empty virtual hard disk
Size (GB)	64
Operating system	Install an operating system from an image file (.iso)
Path	Path to the ISO file you copied to the C:\ClusterStorage\VMStorage volume in the previous

21.1.5 Task 5: Configure virtual machines

1. In the browser window displaying the Windows Admin Center interface, on the **New virtual machine** panel, on the **Inventory** tab, select **vm-000**.
2. On the **Virtual machines > vm-000** panel, identify the **Host** where the VM is located.
3. Within the console session to the **SDNExpress2019-Management** VM, switch to the Server Manager window, select the **Tools** menu, and start **Hyper-V Manager**.
4. In the **Hyper-V Manager** window, right-click or access the context menu for the **Hyper-V Manager** node, and then select **Connect to Server**.
5. In the **Select Computer** dialog box, in the **Another computer** text box, enter the name of the Hyper-V host you identified earlier in this task, and select **OK**.
6. On the **tree** pane of the **Hyper-V Manager** console, select the newly added host. On the **details** pane, right-click or access the context menu for the entry representing the **vm-000** virtual machine, and then select **Connect**.
7. Within the **Virtual Machine Connection** window, select the **Start** button, and when prompted, select any key to initiate the boot process.
8. On the **Windows Server 2019** page, in the **Windows Setup** window, accept the default settings, select **Next**, and then select **Install now**.
9. On the **Select the operating system you want to install** page, select **Windows Server 2019 Datacenter Evaluation**, and then select **Next**.
10. On the **Applicable notices and license terms** page, select the **I accept the license terms** check box, and then select **Next**.
11. On the **Which type of installation do you want?** page, select the **Custom: Install Windows only (advanced)** option.
12. On the **where do you want to install Windows?** page, accept the default settings, and then select **Next**.
13. Switch back to the **Hyper-V Manager** console. In the **details** pane, right-click or access the context menu for the entry representing the **vm-001** virtual machine, and then select **Connect**.
14. Repeat the same sequence of steps you applied to **vm-000** to start installation of **Windows Server 2019 Datacenter Evaluation** on **vm-001** and **vm-100**.

    Note: Wait for the operating installation to complete on all three VMs.

15. Switch back to the **Virtual Machine Connection** window to **vm-000**. When prompted to change the password of the built-in Administrator account, select the **Enter** key, at the **New password** prompt, enter **Pa55w.rd**, select the **Tab** key, enter **Password** again, and then select the **Enter** key.
16. Repeat the previous step to set the password of the built-in Administrator account to **Pa55w.rd** on **vm-001** and **vm-100**.
17. In the browser window, navigate back to the `sddc01.corp.contoso.com` blade, in the list of **Tools**, select **Virtual machines**, and select the **Inventory** tab.
18. On the **Inventory** tab, select the check box next to each virtual machine entry, select **More**, and in the drop-down menu, select **Shut down**, and then select **Yes**.

19. On the **Inventory** tab, select the check next to **vm-000**, select **Settings**, and on the **Settings for vm000** panel, select **Networks**.

20. On the **Networks** panel, specify the following settings:

    Table 10: vm-000 network adapter settings

Setting	Value
Connect to	Virtual network
Virtual network	vnet-000
Virtual subnet	subnet-0 [192.168.0.0/24]
IP Address	192.168.0.100

21. On the **Settings for vm-000** panel, select **Advanced**, on the **Advanced network adapter settings for vm-000**, in the **MAC address type** section, select the **Static** option, and then select **Save**.

    Note: Network Controller automatically assigns the next available MAC address from its pool.

22. Switch to the **Settings for vm-000** panel, and then select **Save network settings**.

23. Repeat the same sequence of steps to assign the following IP configuration to the **vm-001** virtual machine:

    Table 11: vm-001 network adapter settings

Setting	Value
Connect to	Virtual network
Virtual network	vnet-000
Virtual subnet	subnet-1 [192.168.1.0/24]
IP Address	192.168.1.100
MAC address type	Static

24. Repeat the same sequence of steps to assign the following IP configuration to the **vm-100** virtual machine:

    Table 12: vm-100 network adapter settings

Setting	Value
Connect to	Virtual network
Virtual network	vnet-100
Virtual subnet	subnet-0 [192.168.100.0/24]
IP Address	192.168.100.100
MAC address type	Static

21.1.6 Task 6: Test network connectivity of virtual machines

1. On the **SDNExpress2019-Management** VM, in the Windows Admin Center, navigate to the **Inventory** tab of the **Virtual machines** panel of `sddc01.corp.contoso.com` page, select the check boxes next to all three VMs you configured in this task, select **More**, and then in the drop-down menu, select **Start**.

2. Switch to the **Virtual Machine Connection** to **vm-000**, in the **Virtual Machine Connection** window to **vm-000**, in the **Action** menu, select the **Ctrl+Alt+Delete** entry. When prompted, enter **Pa55w.rd**, and then select **Enter**.

3. In the **Virtual Machine Connection** window to **vm-000**, at the Command Prompt, run the following command to disable Windows Defender Firewall:

```
powershell Set-NetFirewallProfile -Profile Domain,Public,Private -Enabled False
```

4. Repeat the previous two steps to disable Windows Defender Firewall on **vm-001** and **vm-100**.

5. Switch back to the **Virtual Machine Connection** window to **vm-000**. From the Command Prompt, run the following command to test connectivity over WinRM to **vm-001**:

```
powershell Test-NetConnection -ComputerName 192.168.1.100 -Port 5985 -InformationLevel Detailed
```

6. Review the output and verify that the connection was successful.

   > **Note**: This output is expected because Remote Management is by default enabled and **vm-000** and **vm-001** are on the same virtual network. The fact that they are not two different subnets does not have any significance in this case.

7. From the Command Prompt, run the following command to test connectivity over WinRM to **vm-100**:

   ```
   powershell Test-NetConnection -ComputerName 192.168.100.100 -Port 5985 -InformationLevel Detailed
   ```

8. Review the output and verify that the connection failed.

   > **Note**: This output is expected because while Remote Management is enabled by default, **vm-000** and **vm-100** are on different virtual networks. At this point, these virtual networks are not connected to each other.

21.1.7 Task 7: Connect virtual networks

1. Switch back to the **SDNExpress2019-Management** VM. In Windows Admin Center, navigate to the **Inventory** tab of the **Virtual networks** panel of the `sddc01.corp.contoso.com` page.
2. On the **Inventory** tab of the **Virtual networks** panel, select the **vnet-000** entry, and then select **Settings**.
3. On the **vnet-000 Settings** panel, select **Peerings**, and on the **Peerings** panel, select **+ New**
4. On the **New Peering** panel, specify the following settings, and then select **Submit**:

   Table 13: vnet-000-to-vnet-100 peering settings

Setting	Value
Name	vnet-000-to-vnet-100
Virtual networks	vnet-100
Allow Virtual network access from ”vnet-000” to remote virtual network	Enabled
Allow forwarded traffic from ”vnet-000” to remote virtual network	Enabled
Allow Gateway Transit	Disabled

5. Switch to the **vnet-000 Settings Peerings** panel, verify that the new peering is listed as **Initiated** or **Connected** in the **Peering Status** column.
6. Navigate back to the **Inventory** tab of the **Virtual networks** panel, select the **vnet-100** entry, and then select **Settings**.
7. On the **vnet-100 Settings** panel, select **Peerings**, and on the **Peerings** panel, select **+ New**.
8. On the **New Peering** panel, specify the following settings and select **Submit**:

   Table 14: vnet-100-to-vnet-000 peering settings

Setting	Value
Name	vnet-100-to-vnet-000
Virtual networks	vnet-000
Allow Virtual network access from ’vnet-100’ to remote virtual network	Enabled
Allow forwarded traffic from ’vnet-100’ to remote virtual network	Enabled
Allow Gateway Transit	Disabled

9. Switch to the **vnet-100 Settings Peerings** panel, and then verify that the new peering is listed as **Connected** in the **Peering Status** column.

21.1.8 Task 8: Test connectivity between peered virtual networks

> **Note**: For the change to take effect, the Network Controller Host agent on the Hyper-V host where the virtual machines reside must process the corresponding policy. To expedite the change, you will restart the agent and each of the virtual machines.

1. Within the console session to the **SDNExpress2019-Management** VM, switch to the browser window displaying the Windows Admin Center, and on the upper left side of the page select **Windows Admin Center**.
2. Select the the Hyper-V host **HV3** to which you deployed all three VMs.
3. On the page displaying the properties of the Hyper-V host, in the **Tools** list, select **Services**.
4. On the **Services** panel, locate and select the **NcHostAgent** entry, and then select **Restart** in the toolbar.
5. Within the console session to the **SDNExpress2019-Management** VM, switch to the **Hyper-V Manager** console displaying the Hyper-V host (**HV3**) to which you deployed all three VMs.
6. In the **Hyper-V Manager** window, select **vm-000**, **vm-001**, and **vm-100**. In the **Actions** pane, in the **Selected Virtual Machines** section, select **Shut down** twice.
7. Wait until all VMs are listed in the **Off** state. In the **Actions** pane, in the **Selected Virtual Machines** section, select **Start**.
8. Switch to the **Virtual Machine Connection** to **vm-000**.
9. In the **Virtual Machine Connection** to **vm-000** window, in the **Action** menu, select the **Ctrl+Alt+Delete** entry. When prompted, enter **Pa55w.rd**, and then select **Enter**.
10. In the **Virtual Machine Connection** to **vm-000** window, from the Command Prompt, run the following to test connectivity over WinRM to **vm-100**:

    ```
    powershell Test-NetConnection -ComputerName 192.168.100.100 -Port 5985 -InformationLevel Detailed
    ```

11. Review the output and verify that the connection was successful.

    > **Note**: This output is expected because Remote Management is enabled by default. At this point, although **vm-000** and **vm-100** are on different virtual networks, there are peering connections between them.

    > **Note**: If the connection fails, wait a few minutes and try again.

21.2 lab: title: 'Lab C: Implementing SDN Access Control List by using Windows Admin Center' type: 'Answer Key' module: 'Module 4: Planning for and Implementing Azure Stack HCI Networking'

22 Lab C answer key: Implementing SDN Access Control List by using Windows Admin Center

22.1 Exercise 1: Implementing SDN Access Control List by using Windows Admin Center

22.1.1 Task 1: Create an ACL

1. To create an access control list (ACL), Within the **SDNExpress2019-Management** VM, in the Windows Admin Center, on the `sddc01.corp.contoso.com` page, in the list of **Tools**, in the **Networking** section, select **Access control lists**.
2. On the **Access control lists** panel, on the **Inventory** tab, select **+ New**. In the **Access Control List** panel, in the **Name** text box, enter **acl-100**, select the **acl-100** link, and then select **Submit**.
3. On the **Access Control List > acl-100** panel, in the **Access Control Rule** section, select **+ New**.
4. In the **Access Control Rule** section, specify the following settings and then select **Submit**:

    The allow-all access rule settings are:

    - Name: **allow-all**
    - Priority: **1000**
    - Types: **Inbound**
    - Protocol: **All**
    - Source Address Prefix: *****
    - Source Port Range: *****

- Destination Address Prefix: *****
- Destination Port Range: *****
- Action: **Allow**
- Logging: **Enabled**

5. On the **Access Control List > acl-100** panel, in the **Access Control Rule** section, select **+ New**.
6. In the **Access Control Rule** section, specify the following deny-winrm-from-vnet-000-subnet-0 access rule settings:

   - Name: **deny-winrm-from-vnet-000-subnet-0**
   - Priority: **500**
   - Types: **Inbound**
   - Protocol: **TCP**
   - Source Address Prefix: **192.168.0.0/24**
   - Source Port Range: *****
   - Destination Address Prefix: *****
   - Destination Port Range: **5985,5986**
   - Action: **Deny**
   - Logging: **Enabled**

7. Select **Submit** and verify that the rule was created successfully.

22.1.2 Task 2: Assign the ACL to a subnet

1. Within the console session to the **SDNExpress2019-Management** VM, start Windows PowerShell ISE as Administrator. In the Windows PowerShell Integrated Scripting Environment (ISE) window, in the **script** pane, open a new tab, paste the following script, and run it to list the properties of the virtual networks you created earlier in this exercise:

   ```
   Import-Module NetworkController
   $uri = 'https://NCCLUSTER.corp.contoso.com'
   Get-NetworkControllerVirtualNetwork -ConnectionUri $uri
   ```

2. In the Windows PowerShell ISE window, in the **script** pane, open a new tab, paste the following script, and run it to assign the access control list you created in the previous task to the first subnet (**subnet-0**) of the virtual network **vnet-100**:

   ```
   $vnet2 = Get-NetworkControllerVirtualNetwork -ConnectionUri $uri -ResourceId 'vnet-100'
   $acl = Get-NetworkControllerAccessControlList -ConnectionUri $uri -resourceid 'acl-100'
   $vnet2.properties.subnets[0].Properties.AccessControlList = $acl
   $subnet = Get-NetworkControllerVirtualSubnet -VirtualNetworkId $vnet2.ResourceId -ConnectionUri $u
   New-NetworkControllerVirtualSubnet -ConnectionUri $uri -Properties $vnet2.Properties.Subnets[0].Pr
   ```

   > **Note**: Verify that the access control list assignment was created successfully.

3. Switch to the browser window displaying the Windows Admin Center and refresh the browser page displaying the **Access control lists > acl-100** panel.
4. On the **Access Control List > acl-000** panel, in the **Related Tab** section, on the **Applied Virtual Subnets** tab, note the **subnet-0** entry of the **vnet-100** virtual network.

22.1.3 Task 3: Verify functionality of the access control list

> **Note**: For the change to take effect, the Network Controller Host agent on the Hyper-V host where the virtual machines reside must process the corresponding policy. To expedite the change, you will restart the agent and the third virtual machine **vm-100**.

1. Within the console session to the **SDNExpress2019-Management** VM, switch to the browser window displaying the Windows Admin Center. On the upper left hand side of the page, select **Windows Admin Center**.
2. Select the the Hyper-V host **HV3** to which you deployed all three virtual machines.
3. On the page displaying the properties of the Hyper-V host, in the **Tools** list, select **Services**.
4. On the **Services** panel, locate and select the **NcHostAgent** entry, and then select **Restart** in the toolbar.

5. Within the console session to the **SDNExpress2019-Management** VM, switch to the **Hyper-V Manager** console displaying the Hyper-V host (**HV3**) to which you deployed all three virtual machines.

6. In the **Hyper-V Manager** window, select **vm-100** and, in the **Actions** pane, in the **vm-100** section, select **Shut down**.

7. Wait until **vm-100** is listed in the **Off** state and, in the **Actions** pane, in the **vm-100** section, select **Start**.

8. Switch to the **Virtual Machine Connection** to **vm-000**. In the **Virtual Machine Connection** window to **vm-000**, from the Command Prompt, run the following command to test connectivity over WinRM to **vm-100**:

   ```
   powershell Test-NetConnection -ComputerName 192.168.100.100 -Port 5985 -InformationLevel Detailed
   ```

9. Review the output and verify that the connection failed.

   > **Note**: This output is expected because Windows Remote Management traffic from **subnet-0** of **vnet-000** to which **vm-000** is attached is blocked by the newly created access control list assigned to **subnet-0** of **vnet-100**, to which the **vm-100** is attached.

10. In the Virtual Machine Connection window to **vm-000**, from the Command Prompt, run the following command to test connectivity over Internet Control Message Protocol (ICMP) to **vm-100**:

    ```
    ping 192.168.100.100
    ```

11. Review the output and verify that the connection was successful.

    > **Note**: This output is expected because all other types of traffic, except for Remote Management, from **vnet-000** (including **subnet-0** to which **vm-000** is attached) is allowed by the newly created access control list (ACL) assigned to **subnet-0** of **vnet-100**, to which the **vm-100** is attached.

12. Switch to the **Virtual Machine Connection** to **vm-001**. In the **Virtual Machine Connection** window to **vm-001**, in the **Action** menu, select the **Ctrl+Alt+Delete** entry. When prompted, enter **Pa55w.rd**, and select **Enter**.

13. In the **Virtual Machine Connection** window to **vm-001**, from the Command Prompt, run the following command to test connectivity over WinRM to **vm-100**:

    ```
    powershell Test-NetConnection -ComputerName 192.168.100.100 -Port 5985 -InformationLevel Detailed
    ```

14. Review the output and verify that the connection was successful.

    > **Note**: This output is expected because Windows Remote Management traffic to **subnet-0** of **vnet-100**, to which the **vm-100** is attached is blocked only from **subnet-0** of **vnet-000**, not from **subnet-1** to which **vm-001** is attached.

15. Switch to the **Virtual Machine Connection** window to **vm-100**. In the **Virtual Machine Connection** window to **vm-100**, in the **Action** menu, select the **Ctrl+Alt+Delete** entry. When prompted, enter **Pa55w.rd**, and then select **Enter**.

16. In the **Virtual Machine Connection** window to **vm-100**, from the Command Prompt, run the following command to test connectivity over WinRM to **vm-000**:

    ```
    powershell Test-NetConnection -ComputerName 192.168.0.100 -Port 5985 -InformationLevel Detailed
    ```

17. Review the output and verify that the connection was successful.

    > **Note**: This output is expected because the access control rule blocking Windows Remote Management traffic applies only to inbound traffic from **subnet-0** of **vnet-000**, and does not apply to outbound traffic from **subnet-0** of **vnet-100**, to which the **vm-100** is attached.

22.2 lab: title: 'Lab D: Implementing SDN Software Load Balancing with private virtual IP by using PowerShell' type: 'Answer Key' module: 'Module 4: Planning for and Implementing Azure Stack HCI Networking'

23 Lab D answer key: Implementing SDN Software Load Balancing with private virtual IP by using PowerShell

23.1 Exercise 1: Implementing SDN Software Load Balancing by using Windows Admin Center and Windows PowerShell

23.1.1 Task 1: Review SDN virtual IP logical network configuration

1. Within the **SDNExpress2019-Management** VM, in the Windows Admin Center, on the `sddc01.corp.contoso.com` page, in the list of **Tools**, in the **Networking** section, select **Logical networks**.
2. On the **Logical networks** panel, on the **Inventory** tab, select the **PrivateVIP** entry.
3. On the **Logical networks > Private VIP** panel, review the configuration in the **Logical subnet** section, and note that it contains a single subnet with the IP address space of **10.20.0.0/24**.
4. On the **Logical networks > Private VIP** panel, select the **10.20.0.0_24** entry, review the configuration, and note that it currently has **1** allocated IP address.

   > **Note**: You will use an IP address from that range as a private virtual IP for connection to load-balanced virtual machines (VMs) on the first of the virtual networks you previously created in this lab.

5. Navigate back to the **Logical networks** panel, on the **Inventory** tab, select the **PublicVIP** entry.
6. On the **Logical networks > Public VIP** panel, review the configuration in the **Logical subnet** section, and note that it contains a single subnet with the IP address space of **10.10.0.0/24**.
7. On the **Logical networks > Private VIP** panel, select the **10.10.0.0_24** entry, review the configuration, and note that it currently has **1** allocated IP address.

   > **Note**: You will use an IP address from that range as a public virtual IP for connection to a VM on the second of the virtual networks you previously created in this lab.

23.1.2 Task 2: Install the Web Server role on VMs in a virtual network

1. Switch to the **Virtual Machine Connection** to **vm-000**. In the **Virtual Machine Connection** window to **vm-000**, from the Command Prompt, run the following command to install the Web Server role.

   ```
   powershell Install-WindowsFeature -Name Web-Server
   ```

2. Use the procedure described in the previous step to install the Web server role on **vm-001**.

   > **Note**: Wait for both installations to complete.

3. Switch to the **Virtual Machine Connection** to **vm-100**. In the **Virtual Machine Connection** window to **vm-100**, from the Command Prompt, run the following commands to verify that the installation was successful.

   ```
   powershell Invoke-WebRequest -Uri 192.168.0.100 -UseBasicParsing
   powershell Invoke-WebRequest -Uri 192.168.1.100 -UseBasicParsing
   ```

   > **Note**: Verify that in both cases you are receiving a response including **HTTP/1.1 200 OK**.

23.1.3 Task 3: Configure an SLB private virtual IP address

1. Switch to the console session to the **SDNExpress2019-Management** VM. In the Windows PowerShell Integrated Scripting Environment (ISE) window, in the **script** pane, open a new tab, paste the following script, and run it to create a load balancer object:

   ```
   Import-Module NetworkController
   $uri = 'https://NCCLUSTER.corp.contoso.com'
   ```

```
$LBResourceId = 'lb-000'
$LoadBalancerProperties = New-Object Microsoft.Windows.NetworkController.LoadBalancerProperties
```

2. Within the console session to the **SDNExpress2019-Management** VM, in the Windows PowerShell ISE window, in the **script** pane, open a new tab, paste the following script, and run it to configure the private virtual IP:

```
$vipIP = '10.20.0.100'
$VIPLogicalNetwork = Get-NetworkControllerLogicalNetwork -ConnectionUri $uri -ResourceId 'PrivateV
$FrontEndIPConfig = New-Object Microsoft.Windows.NetworkController.LoadBalancerFrontendIpConfigura
$FrontEndIPConfig.ResourceId = 'lb-000-fe-1'
$FrontEndIPConfig.ResourceRef = "/loadBalancers/$LBResourceId/frontendIPConfigurations/$($FrontEnd

$FrontEndIPConfig.Properties = New-Object Microsoft.Windows.NetworkController.LoadBalancerFrontend
$FrontEndIPConfig.Properties.Subnet = New-Object Microsoft.Windows.NetworkController.Subnet
$FrontEndIPConfig.Properties.Subnet.ResourceRef = $VIPLogicalNetwork.Properties.Subnets[0].Resourc
$FrontEndIPConfig.Properties.PrivateIPAddress = $vipIP
$FrontEndIPConfig.Properties.PrivateIPAllocationMethod = 'Static'
$LoadBalancerProperties.FrontEndIPConfigurations += $FrontEndIPConfig
```

> **Note**: The virtual IP belongs to the IP address range of the subnet of the logical network you identified in the first task of this exercise.

3. Within the console session to the **SDNExpress2019-Management** VM, in the Windows PowerShell ISE window, in the **script** pane, open a new tab, paste the following script, and run it to configure the back-end address pool, which contains the Dynamic IPs assigned to the load-balanced set of VMs:

```
$BackEndAddressPool = New-Object Microsoft.Windows.NetworkController.LoadBalancerBackendAddressPool
$BackEndAddressPool.ResourceId = 'lb-000-be-1'
$BackEndAddressPool.ResourceRef = "/loadBalancers/$LBResourceId/backendAddressPools/$($BackEndAddr
$BackEndAddressPool.Properties = New-Object Microsoft.Windows.NetworkController.LoadBalancerBacken
$LoadBalancerProperties.backendAddressPools += $BackEndAddressPool
```

4. Within the console session to the **SDNExpress2019-Management** VM, in the Windows PowerShell ISE window, in the **script** pane, open a new tab, paste the following script, and run it to define a health probe that the load balancer will use to determine the health state of the back-end pool members:

```
$Probe = New-Object Microsoft.Windows.NetworkController.LoadBalancerProbe
$Probe.ResourceId = 'lb-000-hp-1'
$Probe.ResourceRef = "/loadBalancers/$LBResourceId/Probes/$($Probe.ResourceId)"
$Probe.properties = New-Object Microsoft.Windows.NetworkController.LoadBalancerProbeProperties
$Probe.properties.Protocol = 'HTTP'
$Probe.properties.Port = '80'
$Probe.properties.RequestPath = '/'
$Probe.properties.IntervalInSeconds = 5
$Probe.properties.NumberOfProbes = 5
$LoadBalancerProperties.Probes += $Probe
```

5. Within the console session to the **SDNExpress2019-Management** VM, in the Windows PowerShell ISE window, in the **script** pane, open a new tab, paste the following script, and run it to define a load balancing rule to distribute traffic that arrives at the front-end IP to back-end IPs. In this case, the back-end pool receives Transmission Control Protocol (TCP) traffic to port **80**.

```
$Rule = New-Object Microsoft.Windows.NetworkController.LoadBalancingRule
$Rule.ResourceId = 'web-000'
$Rule.Properties = New-Object Microsoft.Windows.NetworkController.LoadBalancingRuleProperties
$Rule.Properties.FrontEndIPConfigurations += $FrontEndIPConfig
$Rule.Properties.backendaddresspool = $BackEndAddressPool
$Rule.Properties.protocol = 'TCP'
$Rule.Properties.FrontEndPort = 80
$Rule.Properties.BackEndPort = 80
$Rule.Properties.IdleTimeoutInMinutes = 4
$Rule.Properties.Probe = $Probe
$LoadBalancerProperties.loadbalancingRules += $Rule
```

6. Within the console session to the **SDNExpress2019-Management** VM, in the Windows PowerShell ISE window, in the **script** pane, open a new tab, paste the following script, and run it to apply the change by adding the load balancer configuration to Network Controller:

```
$LoadBalancerResource = New-NetworkControllerLoadBalancer -ConnectionUri $uri -ResourceId $LBResou
```

7. Within the console session to the **SDNExpress2019-Management** VM, in the Windows PowerShell ISE window, in the **script** pane, open a new tab, paste the following script, and run it to retrieve the reference to the load balancer object:

```
$lbresourceid = 'lb-000'
$lb = Get-NetworkControllerLoadBalancer -ConnectionUri $uri -ResourceID $LBResourceId -PassInnerEx
```

8. Within the console session to the **SDNExpress2019-Management** VM, in the Windows PowerShell ISE window, in the **script** pane, open a new tab, paste the following script, and run it to identify values of the **ResourceId** properties of network interfaces assigned to the VMs you previously created in the lab:

```
Get-NetworkControllerNetworkInterface -ConnectionUri $uri | Select-Object ResourceId
```

9. Within the console session to the **SDNExpress2019-Management** VM, in the Windows PowerShell ISE window, in the **script** pane, open a new tab, paste the following script, and run it to add the network interface of **vm-000** to the back-end pool of the load balancer **lb-000**:

```
$nic1 = Get-NetworkControllerNetworkInterface -ConnectionUri $uri -ResourceId 'vm-000_Net_Adapter_
$nic1.properties.IpConfigurations[0].properties.LoadBalancerBackendAddressPools += $lb.properties.
New-NetworkControllerNetworkInterface -ConnectionUri $uri -ResourceId 'vm-000_Net_Adapter_0' -prop
```

10. Within the console session to the **SDNExpress2019-Management** VM, in the Windows PowerShell ISE window, in the **script** pane, open a new tab, paste the following script, and run it to add the network interface of **vm-001** to the back-end pool of the load balancer **lb-000**:

```
$nic2 = Get-NetworkControllerNetworkInterface -ConnectionUri $uri -ResourceId vm-001_Net_Adapter_0
$nic2.properties.IpConfigurations[0].properties.LoadBalancerBackendAddressPools += $lb.properties.
New-NetworkControllerNetworkInterface -ConnectionUri $uri -ResourceId 'vm-001_Net_Adapter_0' -prop
```

23.1.4 Task 4: Verify the configuration of the SDN Software Load Balancing with private virtual IP

1. Within the console session to the **SDNExpress2019-Management** VM, switch to the browser window displaying within the Windows Admin Center interface and navigate to the **Logical subnet > 10.20.0.0__24** panel in the **PrivateVIP** section.

2. Refresh the browser page, review the updated configuration, and note that it currently has **2** allocated IP addresses.

3. In the console session to the **SDNExpress2019-Management** VM, open a new browser window.

4. In the browser window, navigate to `http://10.20.0.100`, and verify that you can access the default IIS home page.

5. Within the console session to the **SDNExpress2019-Management** VM, switch to the Windows PowerShell ISE window, and run the following from the **console** pane to identify the BGP router information (hosted on the **SDNExpress2019-DC** VM):

```
Invoke-Command -ComputerName DC -ScriptBlock {Get-BgpRouter}
```

> **Note**: Note that the BGP peers include two Gateway VMs and the two multiplexer (MUX) VMs.

6. In the Windows PowerShell ISE window, and run the following from the **console** pane to identify the BGP route information, with the router hosted on the **SDNExpress2019-DC** VM:

```
Invoke-Command -ComputerName DC -ScriptBlock {Get-BgpRouteInformation}
```

> **Note**: Note that the output includes two routes to the private virtual IP you configured in this exercise (one per MUX) and that each route was learned from the corresponding MUX VM.

23.1.5 Task 5: Configure outbound NAT by using SLB

1. In the console session to the **SDNExpress2019-Management** VM, in the Windows PowerShell ISE window, in the **script** pane, open a new tab, paste the following script, and run it to create a load balancer object:

   ```
   Import-Module NetworkController
   $uri = 'https://NCCLUSTER.corp.contoso.com'
   $LBResourceId = 'lb-nat-outbound-100'
   $LoadBalancerProperties = New-Object Microsoft.Windows.NetworkController.LoadBalancerProperties
   ```

2. Within the console session to the **SDNExpress2019-Management** VM, in the Windows PowerShell ISE window, in the **script** pane, open a new tab, paste the following script, and run it to create the load balancer, its front-end IP, and the back-end pool:

   ```
   $vipIP = '10.10.0.100'
   $vipLogicalNetwork = Get-NetworkControllerLogicalNetwork -ConnectionUri $uri -resourceid 'PublicVI

   $FrontEndIPConfig = new-object Microsoft.Windows.NetworkController.LoadBalancerFrontendIpConfigura
   $FrontEndIPConfig.ResourceId = 'fe-100'
   $FrontEndIPConfig.ResourceRef = "/loadBalancers/$LBResourceId/frontendIPConfigurations/$($FrontEnd
   $FrontEndIPConfig.Properties = new-object Microsoft.Windows.NetworkController.LoadBalancerFrontend
   $FrontEndIPConfig.Properties.Subnet = new-object Microsoft.Windows.NetworkController.Subnet
   $FrontEndIPConfig.Properties.Subnet.ResourceRef = $vipLogicalNetwork.Properties.Subnets[0].Resourc
   $FrontEndIPConfig.Properties.PrivateIPAddress = $vipIP
   $FrontEndIPConfig.Properties.PrivateIPAllocationMethod = 'Static'
   $LoadBalancerProperties.FrontEndIPConfigurations += $FrontEndIPConfig

   $BackEndAddressPool = new-object Microsoft.Windows.NetworkController.LoadBalancerBackendAddressPoo
   $BackEndAddressPool.ResourceId = 'bepool-100'
   $BackEndAddressPool.ResourceRef = "/loadBalancers/$LBResourceId/backendAddressPools/$($BackEndAddre
   $BackEndAddressPool.Properties = new-object Microsoft.Windows.NetworkController.LoadBalancerBacken

   $LoadBalancerProperties.backendAddressPools += $BackEndAddressPool
   ```

   > **Note**: The virtual IP belongs to the IP address range of the subnet of the logical network you identified in the first task of this exercise.

3. Within the console session to the **SDNExpress2019-Management** VM, in the Windows PowerShell ISE window, in the **script** pane, open a new tab, paste the following script, and run it to define the outbound network address translation (NAT) rule:

   ```
   $OutboundNAT = new-object Microsoft.Windows.NetworkController.LoadBalancerOutboundNatRule
   $OutboundNAT.ResourceId = 'outbound-nat-100'

   $OutboundNAT.properties = new-object Microsoft.Windows.NetworkController.LoadBalancerOutboundNatRu
   $OutboundNAT.properties.frontendipconfigurations += $FrontEndIPConfig
   $OutboundNAT.properties.backendaddresspool = $BackEndAddressPool
   $OutboundNAT.properties.protocol = 'ALL'
   ```

4. Within the console session to the **SDNExpress2019-Management** VM, in the Windows PowerShell ISE window, in the **script** pane, open a new tab, paste the following script, and run it to apply the change by adding the load balancer configuration to Network Controller:

   ```
   $LoadBalancerResource = New-NetworkControllerLoadBalancer -ConnectionUri $uri -ResourceId $LBResou
   ```

5. Within the console session to the **SDNExpress2019-Management** VM, in the Windows PowerShell ISE window, in the **script** pane, open a new tab, paste the following script, and run it to retrieve the reference to the load balancer object:

   ```
   $lbresourceid = 'lb-nat-outbound-100'
   $lb = Get-NetworkControllerLoadBalancer -ConnectionUri $uri -ResourceID $LBResourceId -PassInnerExc
   ```

6. Within the console session to the **SDNExpress2019-Management** VM, in the Windows PowerShell ISE window, in the **script** pane, open a new tab, paste the following script, and run it to identify values of the ResourceId properties of network interfaces assigned to the VMs you previously created in the lab:

```
Get-NetworkControllerNetworkInterface -ConnectionUri $uri | Select-Object ResourceId
```

7. Within the console session to the **SDNExpress2019-Management** VM, in the Windows PowerShell ISE window, in the **script** pane, open a new tab, paste the following script, and run it to add the network interface of **vm-100** to the back-end pool of the load balancer **lb-nat-outbound-100**:

```
$nic1 = Get-NetworkControllerNetworkInterface -ConnectionUri $uri -ResourceId 'vm-100_Net_Adapter_(
$nic1.properties.IpConfigurations[0].properties.LoadBalancerBackendAddressPools += $lb.properties.l
New-NetworkControllerNetworkInterface -ConnectionUri $uri -ResourceId 'vm-100_Net_Adapter_0' -prope
```

23.1.6 Task 6: Verify the configuration of the SDN Software Load Balancing outbound NAT

1. Within the console session to the **SDNExpress2019-Management** VM, switch to the browser window displaying within the Windows Admin Center interface and navigate to the **Logical subnet > 10.10.0.0_24** panel in the **PublicVIP** section.
2. Refresh the browser page, review the updated configuration, and note that it currently has **2** allocated IP addresses.
3. Switch to the lab VM and, from the console pane of the Administrator: Windows PowerShell ISE window, run the following to install the Web Server role:

   ```
   Install-WindowsFeature -Name Web-Server
   ```

   > **Note**: Wait for the installation to complete.

4. On the lab VM, from the console pane of the Administrator: Windows PowerShell ISE window, run the following to identify the local IP configuration:

   ```
   Get-NetIPConfiguration
   ```

   > **Note**: Review the output of the cmdlet you ran in the previous step and identify the IP address of its network interace which is **NOT** used for internal NAT. You will need it in this task.

5. On the lab VM, open a new browser window and navigate to the IP address you identified in the previous step and verify that you can access the default IIS home page.

   > **Note**: This is the default web site installed on the lab VM, accessible via its private IP address. Verify that the Windows Firewall on the lab VM is allowing inbound traffic on port 80 for all network profiles.

6. Record the IP address and switch back to the console session to the **SDNExpress2019-Management** VM.
7. Within the console session to the **SDNExpress2019-Management** VM, switch to the **Virtual Machine Connection** window to **vm-100**. From the Command Prompt, run the following command to test connectivity to the public IP address you identified in the previous step. Replace the `[ip_address]` placeholder with the IP address you recorded in the previous step.

   ```
   powershell Invoke-WebRequest -Uri [ip_address] -UseBasicParsing
   ```

   > **Note**: Verify that you receive a response with the **Status Code** of **200**.

8. In the Windows PowerShell ISE window, run the following from the **console** pane to identify the Border Gateway Protocol (BGP) route information, with the router hosted on the **SDNExpress2019-DC** VM:

   ```
   Invoke-Command -ComputerName DC -ScriptBlock {Get-BgpRouteInformation}
   ```

   > **Note**: Note that the output includes two routes to the public virtual IP you configured in this exercise (one per multiplexer [MUX]) and that each route was learned from the corresponding MUX VM.

23.1.7 Task 7: Configure SLB-based traffic forwarding to a VM in a virtual network

1. Within the console session to the **SDNExpress2019-Management** VM, switch to the Windows PowerShell ISE window. In the **script** pane, open a new tab, paste the following script, and run it to create a public IP address object referencing a public virtual IP:

```
$publicIPProperties = new-object Microsoft.Windows.NetworkController.PublicIpAddressProperties
$publicIPProperties.ipaddress = '10.10.0.200'
$publicIPProperties.PublicIPAllocationMethod = 'static'
```

```
$publicIPProperties.IdleTimeoutInMinutes = 4
$publicIP = New-NetworkControllerPublicIpAddress -ResourceId 'vm-100-pip' -Properties $publicIPPro
```

2. In the Windows PowerShell ISE window, in the **script** pane, open a new tab, paste the following script, and run it to assign the newly created public IP address object to the network interface of the **vm-100** VM:

```
$nic = Get-NetworkControllerNetworkInterface  -connectionuri $uri -resourceid 'vm-100_Net_Adapter_
$nic.properties.IpConfigurations[0].Properties.PublicIPAddress = $publicIP
New-NetworkControllerNetworkInterface -ConnectionUri $uri -ResourceId $nic.ResourceId -Properties
```

23.1.8 Task 8: Verify connectivity to the VM in a virtualnetwork via a public virtual IP.

1. In the Windows PowerShell ISE window, run the following from the **console** pane to identify the BGP route information, with the router hosted on the **SDNExpress2019-DC** VM:

```
Invoke-Command -ComputerName DC -ScriptBlock {Get-BgpRouteInformation}
```

> **Note**: Note that the output includes two routes to the public virtual IP you configured in this exercise (one per MUX) and that each route was learned from the corresponding MUX VM.
>
> **Note**: If the routes are not displayed yet, you might need to wait a few minutes.

2. Within the console session to the **SDNExpress2019-Management** VM, in the Windows PowerShell ISE window, from the **console** pane, run the following to determine whether you have connectivity to the **vm-100** VM through the IP address allocated from the **PublicVIP** subnet.

```
Test-NetConnection -ComputerName 10.10.0.200 -Port 5985 -InformationLevel Detailed
```

3. Verify that the connection attempt was successful.

4. Within the console session to the **SDNExpress2019-Management** VM, in the Windows PowerShell ISE window, from the **console** pane, run the following script to establish a PowerShell Remoting session to the **vm-100** VM via the IP address allocated from the **PublicVIP** subnet.

```
$username = 'Administrator'
$password = ConvertTo-SecureString -String 'Pa55w.rd' -AsPlainText -Force
$creds = New-Object -TypeName System.Management.Automation.PSCredential -ArgumentList $username,$p
Enter-PSSession -ComputerName 10.10.0.200 -Credential $creds
```

5. After the session is established, in the Windows PowerShell ISE window, from the **console** pane, from the `[10.10.0.200]: PS C:\Users\Administrator\Documents>` prompt, run `ipconfig` and verify that the output displays the IP configuration of the **vm-100** VM, with the IP address of **192.168.100.100**.

23.1.9 Task 9: Deprovision the lab resources

1. Within the Remote Desktop session to lab VM, start Windows PowerShell ISE as Administrator.

2. In the PowerShell ISE window, open and run the **F:\WSLab-master\Scripts\Cleanup.ps1** script to remove all VMs provisioned in this lab.