TECHNICAL WHITE PAPER – SEPTEMBER 2021

Successfully Virtualizing Microsoft SQL Server for High Availability on Azure VMware Solutions

BEST PRACTICES GUIDE

Table of contents

Introduction

This document describes the steps and tasks involved in design and configuring a sample Azure VMware solution infrastructure to optimally support a virtualized production Microsoft SQL Server workload. It is intended to provide a representative list of configuration options, architectural guidelines and operational considerations based on the standard practices documented in VMware's published best practices for virtualizing Microsoft SQL Server workloads on the VMware vSphere® platform.

To provide a broader view of these options, this document describes a 3-node cluster of Microsoft SQL Server instances configured as follows:

- A 2-node failover clustering instance (FCI) configuration consisting of two Windows Server 2019 VMs hosting an instance of Microsoft SQL Server and configured for high availability.
- A 2-node always on availability group (AG) consisting of two Windows Server 2019 VMs hosting a high-availability configuration of a database.
- One of the VMs in both scenarios described above is simultaneously participating as a Node in each of the configurations.

Always refer to the following architectural and best practices guides for the latest and most comprehensive information when designing your VMware vSphere-based hybrid cloud infrastructure. These documents (individually and collectively) will remain the authoritative sources for information related to running Microsoft SQL Server workloads on the vSphere platform and any hybrid- or public-cloud based derivative:

– *SQL Server on vSphere Best Practices Guide*
– *Planning Highly Available, Mission Critical SQL Server Deployments with VMware vSphere*
– *Considerations for running Microsoft SQL server workloads on VMware vSAN*
– *Architecting Microsoft SQL Server on VMware vSphere — Best Practices Guide*

Target Audience

This document assumes a knowledge and understanding of VMware vSphere and Microsoft SQL Server. Other than in passing, this document does not describe features, architecture, management or administration of VMware vSphere or Microsoft's Azure Platform and components. Where necessary, this document includes links to existing references and screenshots for the purpose of illustration only.

Architectural staff can use this document to gain an understanding of how the system will work as a whole as they design and implement various components. Engineers and administrators can use this document as a catalog of technical capabilities. Database administrator (DBA) staff can use this document to gain an understanding of how Microsoft SQL Server might fit into a virtual infrastructure. Management staff and process owners can use this document to help model business processes to take advantage of the savings and operational efficiencies achieved with virtualization and hybrid cloud.

Architectural Guidelines and Operational Considerations for Running SQL Server on Azure VMware Solutions

Azure VMware Solutions (AVS) allows users to create vSphere-based data centers (SDDCs) on AVS. Each deployed SDDC includes VMware ESXi™ hosts, VMware vCenter Server®, VMware vSAN™, VMware NSX® components and other software. The same HTML5-based vSphere client is used to manage a SDDC once deployed. This approach allows seamless migration of demanding application workloads without the need to adopt new toolsets or refactor application components.

The planning and designing phase is very important to ensure that migrations of mission-critical application workloads to Azure VMware Solution complete without negatively impacting application SLAs or affecting the performance, availability, manageability and scalability of the workloads. This document summarizes architectural guidelines which will help enterprises when planning a migration.

Use Case Definition

The following uses cases have been identified as the most frequently employed for SQL Server workloads on Azure VMware Solutions:

- Data center extension, like cloud-bursting and test/dev
- Workloads health and performance
- Workload high availability
- Data center evacuation
- Disaster recovery with site resilience
- Application migration and modernization

Each particular use case, or combination of use cases, influences general solution design and necessitates appropriate requirements gathering. This document focuses on use cases specific to workloads health, performance and high availability. We address the health and performance aspects of our objectives by providing a checklist of the common configuration options and recommendations required to avoid common performance bottlenecks in a typical virtualized Microsoft SQL Server infrastructure. We believe that, instead of repeating the usual exercise of providing performance metrics (which we continue to do in separate documents), it is more beneficial that we provide you with actionable configuration tips and recommendation to make your virtualization experience enjoyable.

Rightsizing

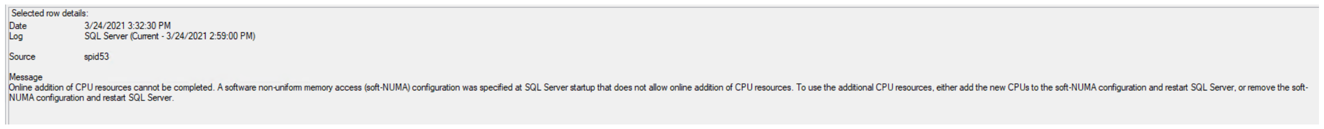
Before considering where to place SQL Server workloads on a cloud platform, ensure that your virtual machine is right-sized. A workload's performance profile should be collected over a sufficient period of time to reflect applications steady state, peaks and spikes in resource utilizations. While defining the required time range to collect time series data, consult with DBAs and application owners to understand the workload profile. At least a full month of "non-rolled up" time series data is recommended prior to executing the performance analysis.

Utilizing the *VMware vRealize® Suite Management Pack for Microsoft SQL Server (vmware.com)* is proven to be very helpful in this preparation phase. While analyzing captured data, make sure your rightsizing approach has been agreed upon by administrators, application owners and business owners, and that it encompasses both spikes (i.e., high performance) and average utilization (i.e., higher density).

Microsoft's *AVS Assessment with Azure Migrate* is also a very useful tool to employ at this phase of your planning, sizing and assessment.

The following should be considered while sizing SQL Server workloads:

- For CPU and memory resources allocation, check *Common questions about Azure VMware Solution* to become familiar with the host's compute maximums and to verify the workload will fit and not overcommit host resources.
- Account for differences in physical CPU architectures between your current environment and the *host instances* used in Azure VMware Solution.
- Always size the CPU resource based on the actual workload, as vCPU can be easily added later.
  - Unnecessarily over-allocating compute resources to a VM can have an adverse performance impact on the workloads.
  - Hot-adding CPUs to a VM running Microsoft SQL Server instances is not recommended.



```
Selected row details:
Date            3/24/2021 3:32:30 PM
Log             SQL Server (Current - 3/24/2021 2:59:00 PM)

Source          spid53

Message
Online addition of CPU resources cannot be completed. A software non-uniform memory access (soft-NUMA) configuration was specified at SQL Server startup that does not allow online addition of CPU resources. To use the additional CPU resources, either add the new CPUs to the soft-NUMA configuration and restart SQL Server, or remove the soft-NUMA configuration and restart SQL Server.
```

  - *Enabling vCPU HotAdd creates fake NUMA nodes on Windows (83980) (vmware.com)*
  - *CPU HotAdd for Windows VMs: How BADLY Do You Want It? - Virtualize Applications (vmware.com)*

- Note that the storage layer in Azure VMware Solution is provided by hyperconverged infrastructure (HCI) solutions VMware vSAN. Adding storage will require the addition of compute resources (i.e., hosts) as well. You can use Azure storage services (e.g., storage accounts, table storage, blob storage) to augment storage for workloads running in your Azure VMware Solution environment.

Requirements

A crucial part of a successful migration is collecting business and technical requirements, allowing you to properly design a cloud platform. For guidance, review *Preparing for Azure VMware Solutions* before beginning your requirements gathering.

Business requirements are an important part of the requirements gathering process. Input examples include:

- RTO/RPO targets
- Business SLAs for the applications workloads based on SQL Server databases
- *Licensing considerations*
- *Azure Hybrid Benefit*
- Security and data-management considerations

Technical requirements will directly influence logical design and should be collected and validated with care. Pay special attention to the following bullet points:

- Performance requirements of the workload (e.g., transactions-per-second, number of user connections, expected future workloads changes)
- Capacity requirements (e.g., future growth, other projects to be served)
- Scalability requirements (e.g., method for increasing capacity of a SDDC, scale-out versus scale-in approach)
- Application requirements (e.g., type of workloads [e.g., OLTP/data warehouse], dependencies between on-premises components and network flow between them)
- Manageability requirements (e.g., providing access to a SDDC to appropriate user groups, reconfiguring monitoring tools, backup solution in use, modifying scripting, vRealize Operations workflows)
- Availability requirements (e.g., SQL Server high-availability solutions in use, DRS groups, host-isolation response, number of availability zones required)

NOTE: Manageability and Availability are two important considerations we will discuss in greater details in this document. This is because, due to its hosted and managed-services nature, AVS provides restricted administrative access to the underlying infrastructure. This level of access limits certain configurations that require assisted support and which cannot ordinarily be performed by the customer.

Risks, Assumptions and Constraints

Ensure that risks, assumptions and constraints are identified and documented and that the risk-mitigation plan has been agreed by all groups involved.

- An example of an assumption: Do you have a software assurance agreement for Microsoft SQL Server licensing?
- An example of a constraint: available network bandwidth between on-premises and SDDC (e.g., is ExpressRoute available?)
  - The scenario described in this document assumes that there is connectivity between the on-premises VMware vSphere infrastructure and the Azure VMware Solution environment
  - The examples shown in this document present a stretched-network scenario in which one network subnet is in use across both environments.
- An example of a risk: different CPU generations between on-premises ESXi hosts and hosts in an SDDC (can you enable EVC to ensure migrating running workloads from on-premises to AVS without impacting service?)

High-Level Architecture

A high-level solution architecture should include enough information to capture the on-premises environment hosting the SQL Server workloads and planned SDDC(s) with multiple availability zones (AZ) while also providing enough details to work on logical design.

Deploying and operating an Azure VMware Solution SDDC involves several simplified processes which are described in the following quick steps:

- *Planning the Deployment*
- *Deploying the Azure VMware Solution*
- *Connecting to on-premises*
- *Deploying and Configuring VMware HCX*

Logical Design

A logical design describes all technical decisions made and addresses identified technical requirements while minimizing risk. The level of detail included should be sufficient to create an implementation guide for the solution. While specifics of each logical design are unique, it's important to ensure all technical prerequisites are met. The following prerequisites have been identified as crucial to the successful migration of SQL Server workloads to Azure VMware Solutions:

- For on-premises-located VMs, ESXi hosts, and/or vSphere clusters hosting SQL server workloads, check and document all advanced settings configured. Ensure corresponding options are available in Azure VMware Solutions. For example, DRS anti-affinity groups and rules must be re-created in an SDDC as they cannot be migrated.

  IMPORTANT:

  a. As at the time of this writing, VM-VM anti-affinity rule is not supported in AVS
  b. As at the time of this writing, creating the necessary anti-affinity rules required to separate clustered VMs in AVS requires that you open a support request with the Microsoft AVS support team. This task is assumed to have been completed and is, therefore, not covered in this document.
- Conduct an assessment of your current infrastructure using *AVS Assessment with Azure Migrate* to ensure that all requirements are met.
- Determine whether or not *Azure Migrate* is appropriate as the migration tool for existing moving workloads to AVS.
- Where applicable, *Deploy and Configure VMware HCX for Azure VMware Solution* to enable you to move existing workloads from your existing VMware infrastructure to AVS.

  a. Hybrid coexistence using L2 VPN between the on-premises and the SDDC is currently not supported for AVS.
- Configure *Hybrid Linked Mode* to allow managing both on-premises and public SDDCs within a single vSphere Client interface. Verify that all required user accounts are added to the appropriate group.

Operational Considerations

Post-implementation maintenance and operation guidelines are a key component of any well-prepared infrastructure architecture. While incorporating Azure VMware Solution SDDCs in an existing infrastructure, it's critical that *Day 2* operational routines are updated accordingly, including:

- Backup configurations
- Monitoring configurations
- Operational documentations

If *vRealize Operations* is used to monitor the environment, confirm that all SDDCs are added to vRealize Operations-managed resources and configured using vCenter Adapter with the **Public Cloud** option.

Configuration Checklist for Hosting Microsoft SQL Server Workloads on AVS

The purpose of this section is to document the specific configuration options for successfully and optimally hosting Microsoft SQL Server instances on the Azure VMware Solution. These are explained in detail in the SQL Server on *vSphere Virtualization Best Practices Guide* mentioned at the beginning of this document. We will therefore not discuss this in great detail.

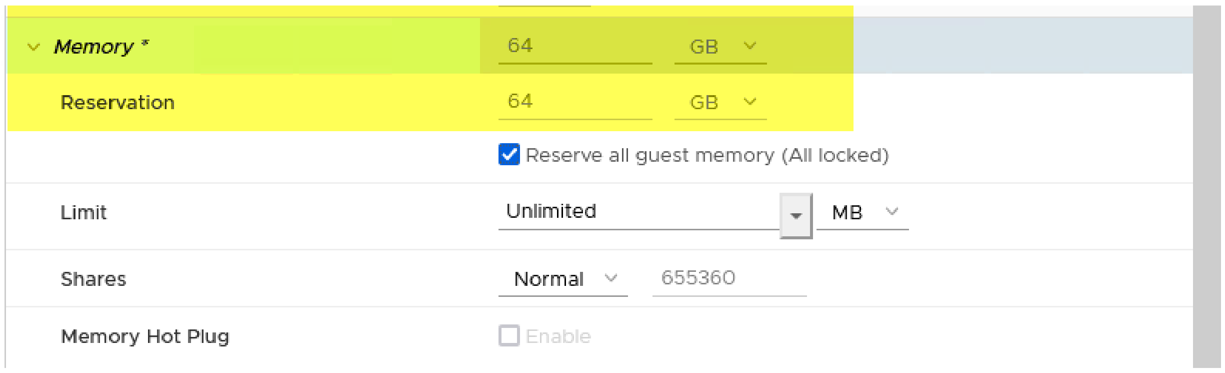
Memory Allocation

How much memory to allocate to a VM on the vSphere platform is strictly a function of the actual utilization threshold, as determined through administrative due diligence. While Microsoft SQL Server Administrators typically prefer to allocate the maximum possible RAM to their Microsoft SQL Server workloads (in anticipation of peak or worst-case consumption scenarios), it is very important to resist this practice. Virtualization imposes certain optimization and functionality considerations that may (counter-intuitively) induce performance penalties for an over-sized VM if over-allocation of resources results in sustained resource wastage. Right-sizing VMs is a sound administrative practice and we encourage applications and virtualization team members to engage in the necessary due diligence to avoid allocating compute resources to a VM on a whim. Memory should be allocated to a Microsoft SQL Server VM strictly based on empirical and historical usage trends.

Memory Hot-Add: Memory hot-add is a feature which allows Administrators to allocate just-enough memory to a VM (without incurring over-allocation penalties) and to adjust this allocation upwards when load increases without having to restart the VM or interrupt running tasks and processes. Modern Windows Server OS, Microsoft SQL Server and VMware vSphere support this feature, so we recommend that, when in doubt, it should be enabled as a compromise, rather than the alternative option of anticipatory over-allocation.

Memory Reservation

One of the tenets and benefits of virtualization is the cost-beneficial opportunity to utilize physical compute resources more densely and fully than is otherwise possible in a physical server environment. At a very high level, the cost benefits come from the ability to pool these resources together and share them more broadly among more OS instances on one physical server.


Memory *	64	GB
Reservation	64	GB
	☑ Reserve all guest memory (All locked)	
Limit	Unlimited	MB
Shares	Normal	655360
Memory Hot Plug	☐ Enable	

Conversely, one of the challenges of virtualization is the likelihood that, in attempting to share pooled resources among as many OS instances as possible to achieve this density, Administrators run the risk of unintentionally creating the opposite effect, at least for some of the OS instances sharing the pooled resources.

As an example, imagine an ESXi host with 128GB of RAM and 10 VMs, each with 20GB RAM allocated. In normal operating conditions, it is unlikely that all 10 VMs will need their 20GB allocation at the same time. In that scenario, the VMs can all happily coexist without inducing performance bottleneck.

What if all VMs were to suddenly need their full allocation, though? Obviously, we are 72GB of memory short. Since virtualization itself does not manufacture non-existent compute resources, the hypervisor has to resort to various mechanisms (e.g., swapping, ballooning) to try to satisfy the VM’s demands. In the process, some (if not all) of the VMs and the applications they host will experience performance degradation, leading to negative impact on business processes and overall user experience.

Here is where the Memory Reservation feature of VMware vSphere can come into play. In most network environments, workloads are not created equal. Some servers (physical or VMs) host applications which are more critical to business operations and revenue than others. These are called the Business (or Mission) Critical Applications. Supposing that five of the VMs in our example above are considered *"Business Critical"* (aka Tier 1) and the other five are considered not so critical. Memory Reservation would allow us to give each of the five Tier-1 VMs all their allocated memory even as the other five gasp for air.

If you are mixing VMs of different critical importance in your AVS environment, and you anticipate over-allocating available physical memory resources in the ESXi cluster, it is recommended that you reserve ALL the memories you are allocating to the VMs hosting the most critical application in the mix. By doing this, you ensure that your important VMs will always be entitled (and have access) to the allocated memory resources. As a little bonus, because vSphere won't create a swap file for VMs whose memory have been fully reserved, you also save on storage.

CPU Allocation

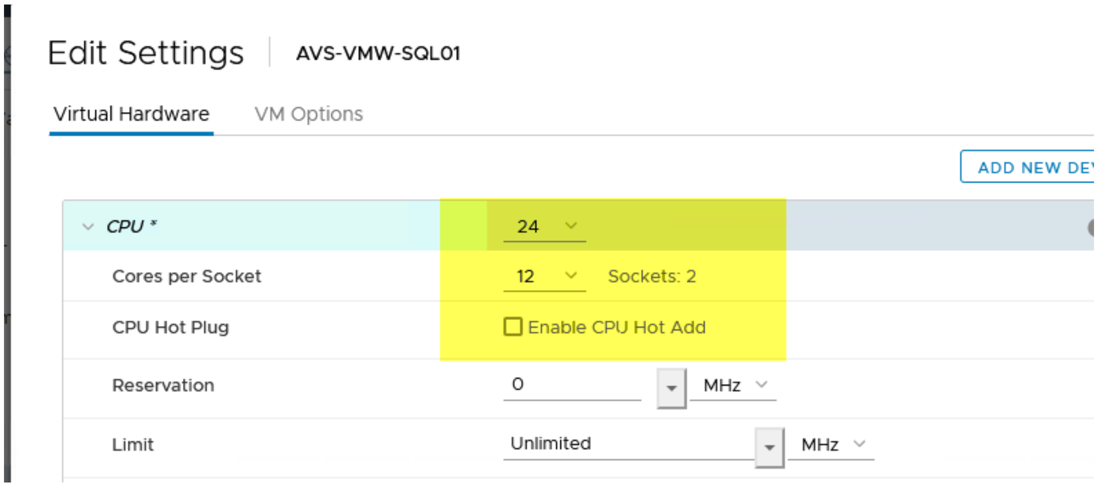
Starting with vSphere 6.5, VMware has changed its prescriptive guidance for allocating compute resources to a VM. Please see *Virtual Machine vCPU and vNUMA Rightsizing - Guidelines - VROOM! Performance Blog (vmware.com)* for all the background and pertinent information around this.

In summary, for wide VMs (i.e., VMs which require more vCPU counts than are available in one physical socket in the ESXi host), the recommendation to *leave cores-per-socket at 1* no longer applies. The simplified rule of thumb is to allocate vCPUs to a VM in a way that mirrors the physical NUMA topology of the ESXI host.

CPU Reservation: This is not recommended, so we will not spend much time discussing this.

CPU Hot-Add: While Microsoft SQL Server is one of the few modern applications which can dynamically adapt to (and utilize) hot-added CPUs, VMware does not recommend enabling hot-add CPUs for VMs running any version of the Windows Server Operating System at this time. Please see the following for more detailed discussion:

– *CPU HotAdd for Windows VMs: How BADLY Do You Want It? - Virtualize Applications (vmware.com)*
– *Enabling vCPU HotAdd creates fake NUMA nodes on Windows (83980) (vmware.com)*
– *CPU Hot Add Performance in vSphere 6.7 - VROOM! Performance Blog (vmware.com)*


Edit Settings | AVS-VMW-SQL01

Virtual Hardware | VM Options

ADD NEW DE

CPU *	24	
Cores per Socket	12	Sockets: 2
CPU Hot Plug	☐ Enable CPU Hot Add	
Reservation	0	MHz
Limit	Unlimited	MHz

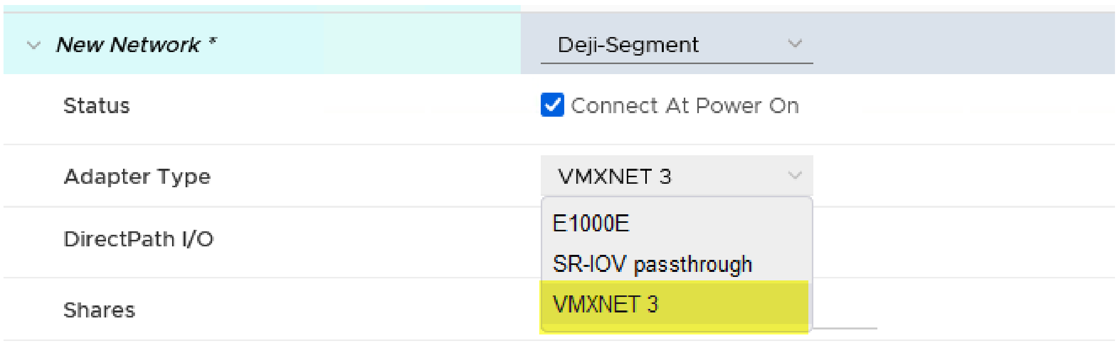
Beware of "Limit"

As you have likely noticed in the previous two images above, vSphere provides Administrators an easy option to prevent unnecessary resource wastage by imposing a limit on the Memory and CPU resources allocated to a VM. VMware strongly recommends that Administrators leave this option unconfigured in both cases. It is better to right-size and allocate as much resources to a Microsoft SQL Server VM as it requires rather than over-allocating resources and then consciously imposing a restriction on the VM's access to the allocated resources.

One of the most challenging problems with configuring Limits on VMs is that such configuration is easily forgotten and then missed during a performance troubleshooting exercise. The standard administrative response when a VM is perceived to be resource-constrained is to increase its allocation. Sadly, however much one increases the allocation, the VM will never be able to utilize anything beyond what has been set in **Limit**.

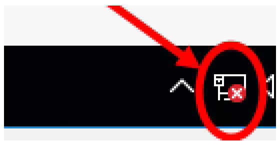
Network Card Configuration

When you create a Windows VM on vSphere, the default vNIC type assigned to the VM is the E1000E vNIC. This is not a suitable vNIC type a Windows VM running Microsoft SQL Server workloads. VMware highly recommends that you always assign the VMXNet3 virtual network card type to any Windows VMs which have more than one vCPU allocated. This is primarily because the VMXNet3 vNIC type is the only vNIC type in vSphere which supports the Windows Receive Side Scaling (RSS) feature. No other vNIC has this capability.



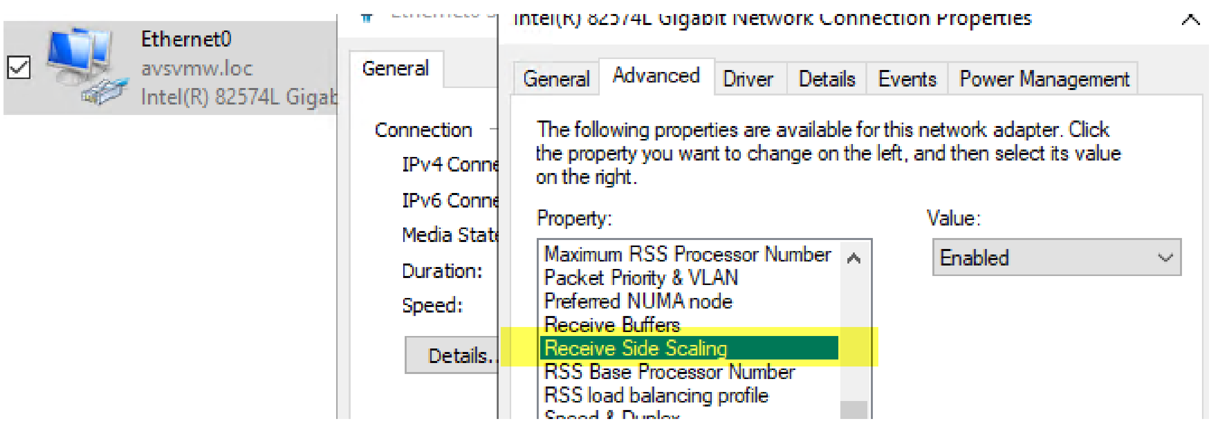
New Network *	Deji-Segment
Status	✓ Connect At Power On
Adapter Type	VMXNET 3
DirectPath I/O	
Shares	

So, why would should you care about RSS for your MS SQL Server VMs? Here is why: *Introduction to Receive Side Scaling - Windows drivers | Microsoft Docs*. Briefly, RSS is the Windows feature which allows Windows to distribute the handling of network packets/requests/interrupts over multiple CPUs in the OS. Without this feature, only one CPU (out of however many you have allocated to Windows) will be doing the heavy lifting. On a busy Windows OS instance, this invariably leads to packet drops, performance degradation and (eventually) instability.



Choosing VMXNet3 vNIC for the VM is just one step. Because VMXNet3 is not native to Windows, it is unusable until you have installed the VMTools on the VM. It is very important that you always keep VMTools up-to-date on your VMs because this is the mechanism through which VMware delivers updated drivers and fixes to the VM and its components.

By default, RSS is not enabled on the vNIC inside Windows, so you must make enabling it a core part of your standard administrative or deployment procedures (better yet, enable it in your Windows VM Templates).


Ethernet0
avsvmw.loc
Intel(R) 82574L Gigab
General
Connection
IPv4 Conne
IPv6 Conne
Media Stat
Duration:
Speed:
Details.

Intel(R) 82574L Gigabit Network Connection Properties

General | Advanced | Driver | Details | Events | Power Management

The following properties are available for this network adapter. Click the property you want to change on the left, and then select its value on the right.

Property:
Maximum RSS Processor Number
Packet Priority & VLAN
Preferred NUMA node
Receive Buffers
Receive Side Scaling
RSS Base Processor Number
RSS load balancing profile
Speed & Duplex

Value:
Enabled

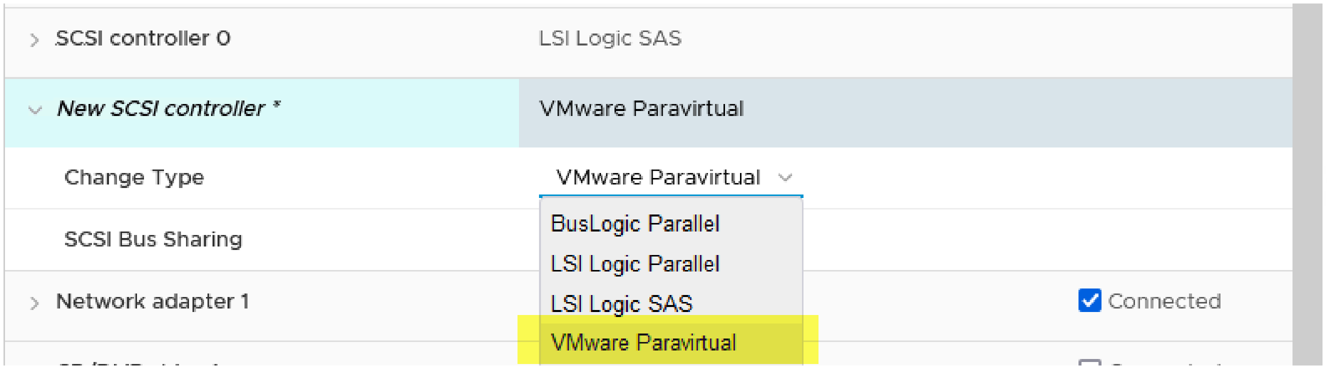
NOTE: Please disregard any counter recommendation you may come across. If it's a Windows VM, and it has multiple processors and it's likely to generate, receive, or transmit network traffic, VMXNet3 is your only choice for virtual network card.

SCSI Controller Configuration

The default virtual SCSI controller type assigned to a Windows VM on vSphere is the LSI Logic SAS. This choice is driven largely by the need to simplify the VM deployment process, allowing the Administrator to install Windows as quickly and seamlessly as possible.

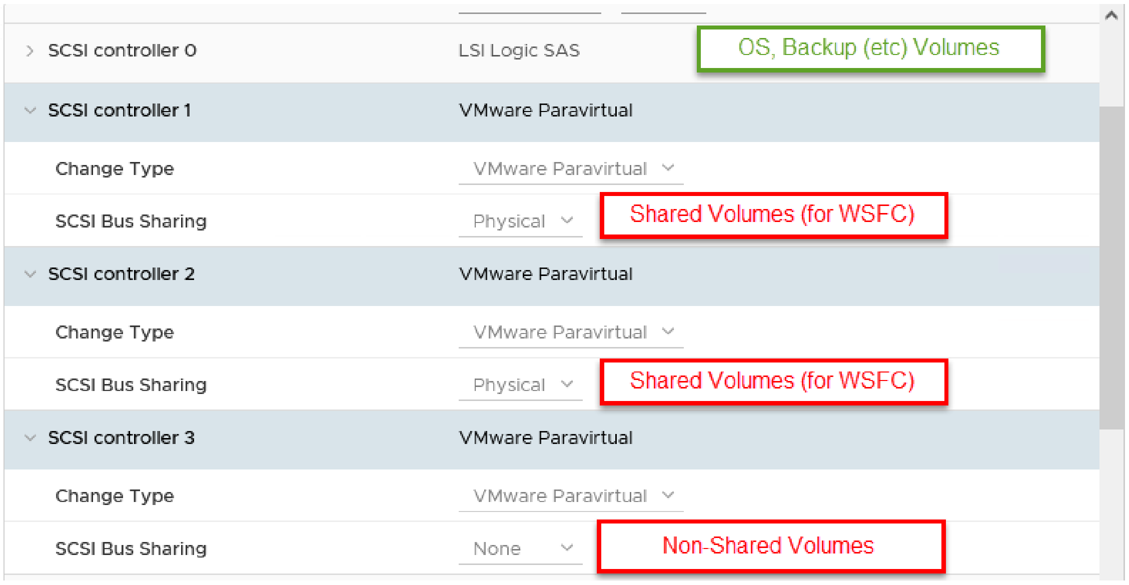
However, the LSI Logic SAS SCSI controller is inadequate for the data throughput and disk IO patterns of a typical Microsoft SQL Server workload, which expects an optimal and fast storage IO send/receive rate for its queries and transactions. The LSI Logic SAS vSCSI controller is limited to 32 IO queue depth, which is insufficient for most modern Microsoft SQL Server workloads.

The VMware Paravirtualized Controller (PVSCSI), on the other hand, has a configurable queue depth of 64, which can be increased to 254 (per device) and 1024 (per Adapter). In addition, PVSCSI has a lighter CPU footprint, which makes it even that much more performant than the default LSI Logic SAS.


SCSI controller 0: LSI Logic SAS
*New SCSI controller **: VMware Paravirtual
Change Type: VMware Paravirtual
SCSI Bus Sharing
Network adapter 1: Connected

BusLogic Parallel
LSI Logic Parallel
LSI Logic SAS
VMware Paravirtual

VMware highly recommends PVSCSI for all Data, Transaction Logs and TempDB disks of a Microsoft SQL Server VM. If you have multiple disks (VMDKs) for these disk types, VMware recommends that customers add additional PVSCSI controllers and distribute the disks as evenly as possible over all the controllers.



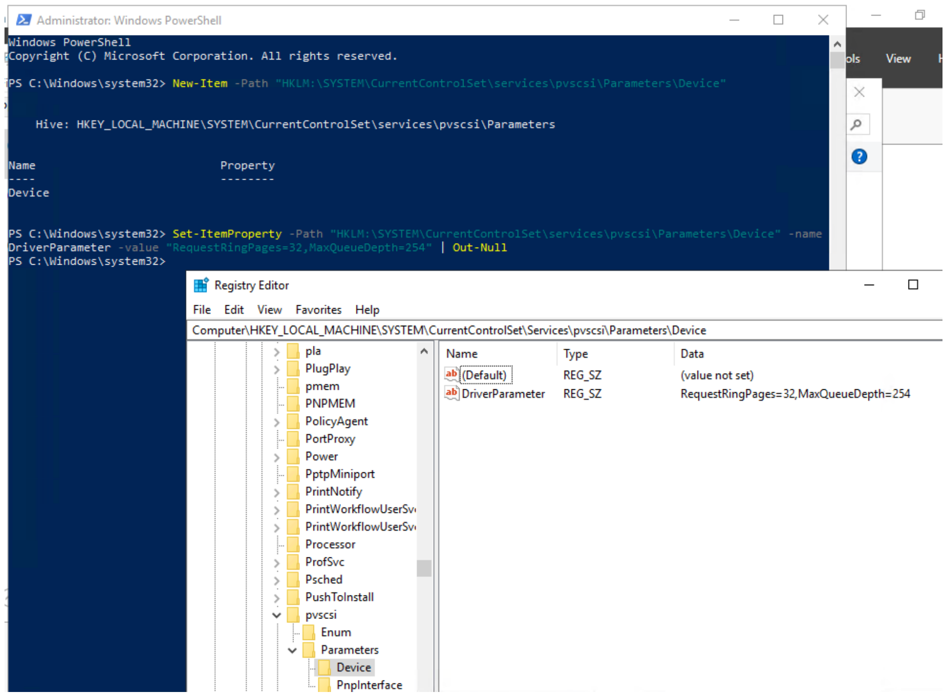
SCSI controller 0	LSI Logic SAS	OS, Backup (etc) Volumes
SCSI controller 1	VMware Paravirtual	
Change Type	VMware Paravirtual	
SCSI Bus Sharing	Physical	Shared Volumes (for WSFC)
SCSI controller 2	VMware Paravirtual	
Change Type	VMware Paravirtual	
SCSI Bus Sharing	Physical	Shared Volumes (for WSFC)
SCSI controller 3	VMware Paravirtual	
Change Type	VMware Paravirtual	
SCSI Bus Sharing	None	Non-Shared Volumes

As with the VMXNet3 vNIC, PVSCSI is not native to Windows and is, therefore, unusable until you have installed VMTools. After installing the VMTools on a Windows VM which has PVSCSI controllers attached, a new Windows Registry key becomes available and configurable. This key is where the controller can be configured to support larger queue depth.

It is highly recommended that customers configure this option in order to increase IO performance for the workloads. The following two-line PowerShell command can be used to accomplish this task:

New-Item -Path "HKLM:\SYSTEM\CurrentControlSet\services\pvscsi\Parameters\Device"

Set-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\services\pvscsi\Parameters\Device" -name DriverParameter -value "RequestRingPages=32,MaxQueueDepth=254" | Out-Null



Administrator: Windows PowerShell

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Windows\system32> New-Item -Path "HKLM:\SYSTEM\CurrentControlSet\services\pvscsi\Parameters\Device"


    Hive: HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\pvscsi\Parameters


Name                           Property
----                           --------
Device


PS C:\Windows\system32> Set-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\services\pvscsi\Parameters\Device" -name
DriverParameter -value "RequestRingPages=32,MaxQueueDepth=254" | Out-Null
PS C:\Windows\system32>
```

Registry Editor

Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\pvscsi\Parameters\Device

Name	Type	Data
(Default)	REG_SZ	(value not set)
DriverParameter	REG_SZ	RequestRingPages=32,MaxQueueDepth=254

Disk Allocation and Configuration

VMware vSphere allows you to create VMDKs as large as 62TB each. While this is generally a good thing, Administrators should resist the temptation to misuse this capability. As with the SCSI controllers, VMDKs have queue depth limits, too. See *Large-scale workloads with intensive I/O patterns might require queue depths significantly greater than Paravirtual SCSI default values (2053145) (vmware.com)* for additional information.

A device (in this case, a VMDK) has a queue depth of 64. The queue depth controls the maximum parallel IO that could be passed through the device at one time. Once this maximum is reached, IOs begin to queue up. If you create a large (62TB) VMDK, slice it up into multiple volumes inside the Guest OS, and assign these to, say, a few Transaction Logs, TempDB and Data disks, all of the IOs generated by these different disks will have to pass through the one VMDK, increasing your chances of reaching the 64 queue depth threshold very frequently and quickly. The end result is storage IO performance degradation.

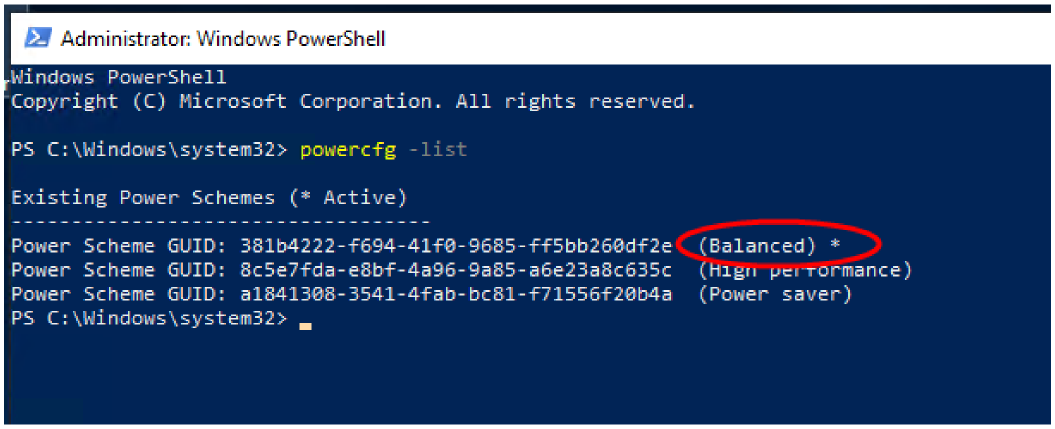
For performance and availability reasons, VMware recommends that Administrators consciously avoid putting too many high-transactional volumes or disks in one VMDK.

What about Eager-Zero Thick disk provisioning? This is an artefact that is not relevant for our purposes.

We have one more important performance-improvement related disk tuning task to check off, but since it is application-specific, *we will discuss it at a later point in this document.*

Power Setting Configuration

Most modern hardware and Operating Systems default to power conservation in consideration of the impact of power consumption on the environment. Unfortunately, conserving power leads to CPU throttling, which leads to severe negative impact on application performance and throughput. In Windows, for example, the default Power Scheme is **Balanced**. Balanced Power Scheme results in Windows putting some CPUs to sleep when it determines that they are not in active use, and then waking them up when it deems it necessary. This behavior invariably results in suboptimal performance for Microsoft SQL Server queries and tasks.



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Windows\system32> powercfg -list

Existing Power Schemes (* Active)
-----------------------------------
Power Scheme GUID: 381b4222-f694-41f0-9685-ff5bb260df2e  (Balanced) *
Power Scheme GUID: 8c5e7fda-e8bf-4a96-9a85-a6e23a8c635c  (High performance)
Power Scheme GUID: a1841308-3541-4fab-bc81-f71556f20b4a  (Power saver)
PS C:\Windows\system32> _
```

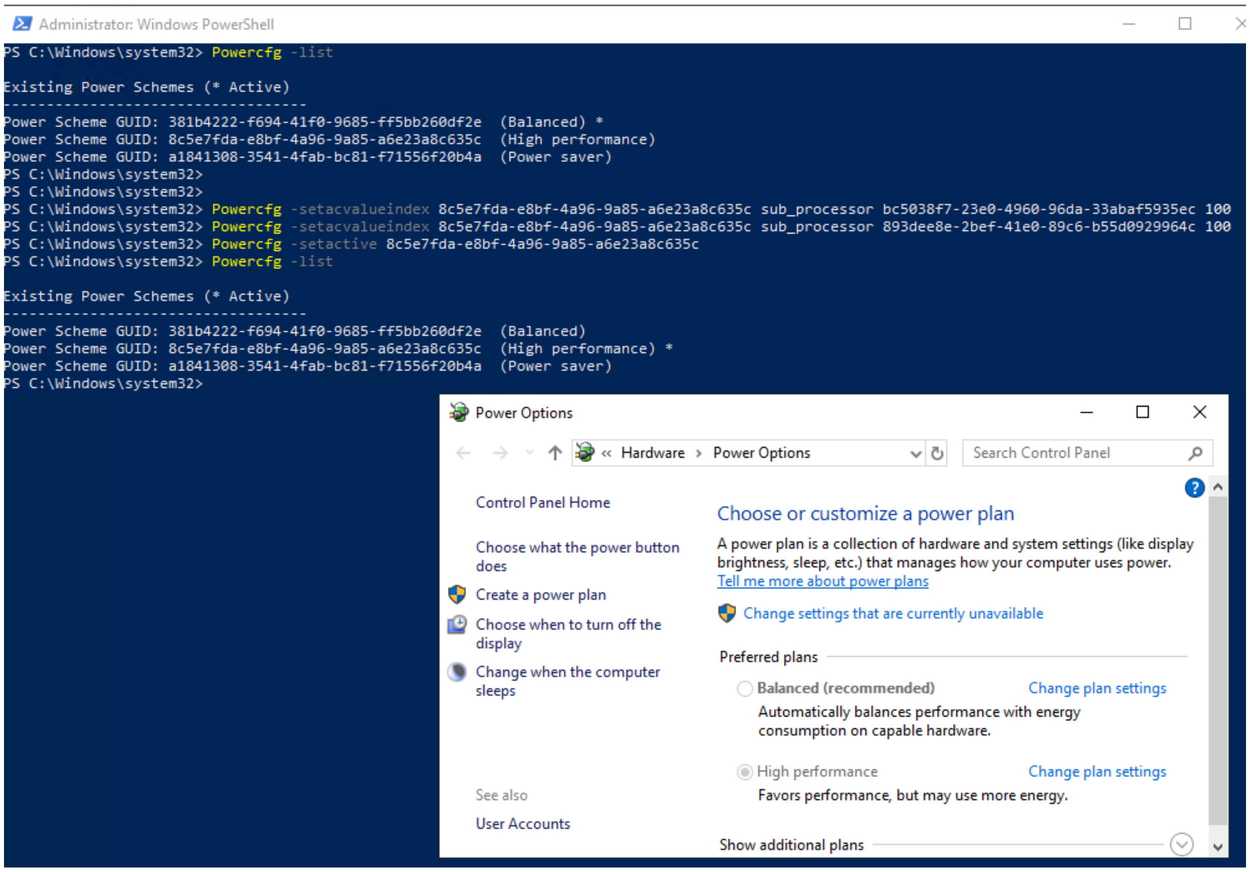
VMware recommends that Administrators change the default **Balanced** power setting to **High Performance** as a standard administrative procedure for all VMs hosting Microsoft SQL Server workloads.

The following PowerShell command will toggle Windows Power Scheme from **Balanced** to **High Performance**:

Powercfg -setacvalueindex 8c5e7fda-e8bf-4a96-9a85-a6e23a8c635c sub_processor bc5038f7-23e0-4960-96da-33abaf5935ec 100

Powercfg -setacvalueindex 8c5e7fda-e8bf-4a96-9a85-a6e23a8c635c sub_processor 893dee8e-2bef-41e0-89c6-b55d0929964c 100

Powercfg -setactive 8c5e7fda-e8bf-4a96-9a85-a6e23a8c635c



Administrator: Windows PowerShell

```
PS C:\Windows\system32> Powercfg -list

Existing Power Schemes (* Active)
-----------------------------------
Power Scheme GUID: 381b4222-f694-41f0-9685-ff5bb260df2e  (Balanced) *
Power Scheme GUID: 8c5e7fda-e8bf-4a96-9a85-a6e23a8c635c  (High performance)
Power Scheme GUID: a1841308-3541-4fab-bc81-f71556f20b4a  (Power saver)
PS C:\Windows\system32>
PS C:\Windows\system32>
PS C:\Windows\system32> Powercfg -setacvalueindex 8c5e7fda-e8bf-4a96-9a85-a6e23a8c635c sub_processor bc5038f7-23e0-4960-96da-33abaf5935ec 100
PS C:\Windows\system32> Powercfg -setacvalueindex 8c5e7fda-e8bf-4a96-9a85-a6e23a8c635c sub_processor 893dee8e-2bef-41e0-89c6-b55d0929964c 100
PS C:\Windows\system32> Powercfg -setactive 8c5e7fda-e8bf-4a96-9a85-a6e23a8c635c
PS C:\Windows\system32> Powercfg -list

Existing Power Schemes (* Active)
-----------------------------------
Power Scheme GUID: 381b4222-f694-41f0-9685-ff5bb260df2e  (Balanced)
Power Scheme GUID: 8c5e7fda-e8bf-4a96-9a85-a6e23a8c635c  (High performance) *
Power Scheme GUID: a1841308-3541-4fab-bc81-f71556f20b4a  (Power saver)
PS C:\Windows\system32>
```

Power Options

« Hardware › Power Options

Search Control Panel

Control Panel Home

Choose what the power button does

Create a power plan

Choose when to turn off the display

Change when the computer sleeps

See also

User Accounts

Choose or customize a power plan

A power plan is a collection of hardware and system settings (like display brightness, sleep, etc.) that manages how your computer uses power.

Tell me more about power plans

Change settings that are currently unavailable

Preferred plans

Balanced (recommended) — Change plan settings

Automatically balances performance with energy consumption on capable hardware.

High performance — Change plan settings

Favors performance, but may use more energy.

Show additional plans

This concludes the high-level performance tuning checklist required to get our VMs ready for an optimal and satisfactory virtualization experience on the Azure VMware Solution platform.

We will now proceed through setting up and configuring a representative use-case scenario, showing how to successfully deploy and configure Microsoft SQL Server workloads to satisfy high-availability requirements.

The configuration shown in the following example is highly simplified. This is intentionally so because we want to focus on the specific considerations for achieving our objectives solely from a virtualization perspective as opposed to offering a tutorial on how to install, configure or cluster Microsoft SQL Server. We assume that you are already familiar with most of the tasks involved in installing, running and operating Microsoft SQL Server.

For specific Application-level performance tuning option, please see *SQL Server Configuration* here and in Section 4.3 (SQL Server Configuration) of *SQL-Server-on-VMware-Best-Practices*.

Deploying and Preparing a VM for Windows Server Failover Clustering

Overview

As described earlier, we will use three VMs in our example. They are named:

- AVS-VMW-SQL01
- AVS-VMW-SQL02
- AVS-VMW-SQL03

Two of the VMs (AVS-VMW-SQL01 and AVS-VMW-SQL02) will host a clustered instance of Microsoft SQL Server in Always On failover clustering instance (FCI) mode.

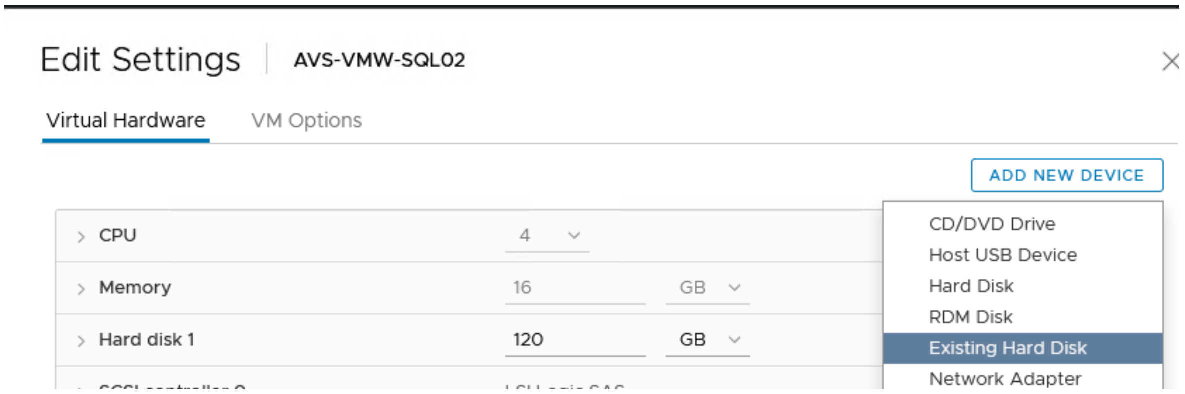
We will also create a Database on this Instance (AVSVMW-DemoAGDB) and configure it for high availability in Always On Availability Group (AG) mode.

AVS-VMW-SQL03 will be configured to host an asynchronous replica of AVSVMW-DemoAGDB.

We'll henceforth refer to these VMs as SQL01, SQL02, and SQL03.

Deployment Steps

- We create the VMs, following the configuration options described earlier in this document.
- We create one or more VMDKs on SQL01 which will be used as shared disks for our FCI. In our simplified example, we're using only one shared VMDK.
- We attach this VMDK to Bus 0 on PVSCSI Controller 1 (SCSI1:0). Remember that the SCSI bus-sharing mode has to be set to **Physical** when used for shared disk.
- We are done with SQL01 now, so we move on to SQL02
- On SQL02, we add this VMDK we just configured on SQL01 to SQL02. We do this by selecting the **Existing Hard Disk** option in the **Add New Device** menu.

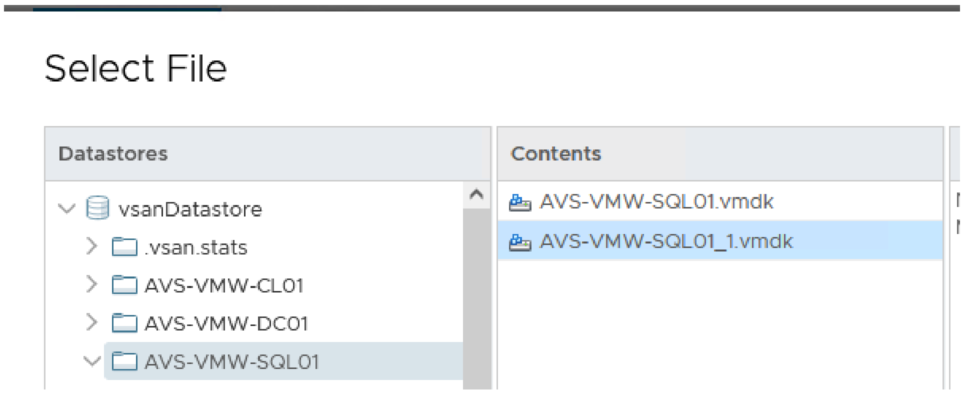

Edit Settings | AVS-VMW-SQL02

Virtual Hardware | VM Options

ADD NEW DEVICE

CPU	4	
Memory	16	GB
Hard disk 1	120	GB

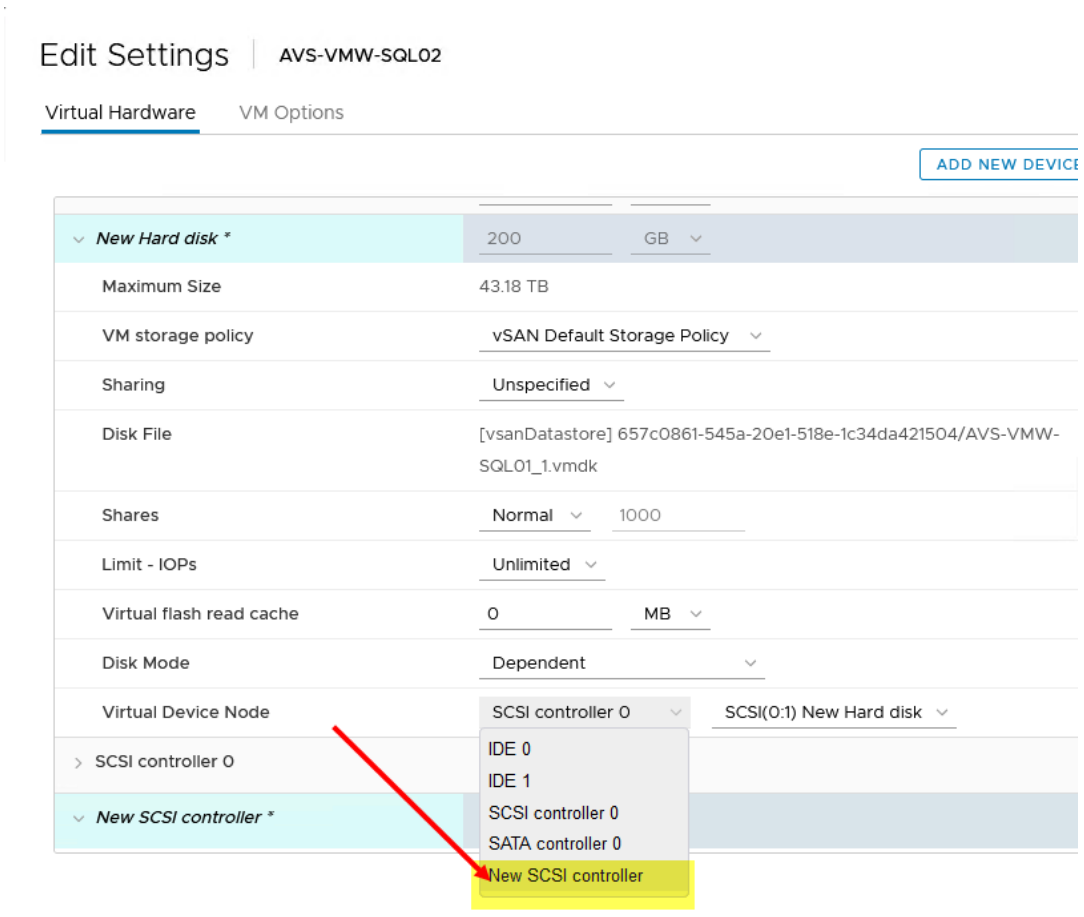
• We browse to where we have stored the shared VMDK and select it.



Select File

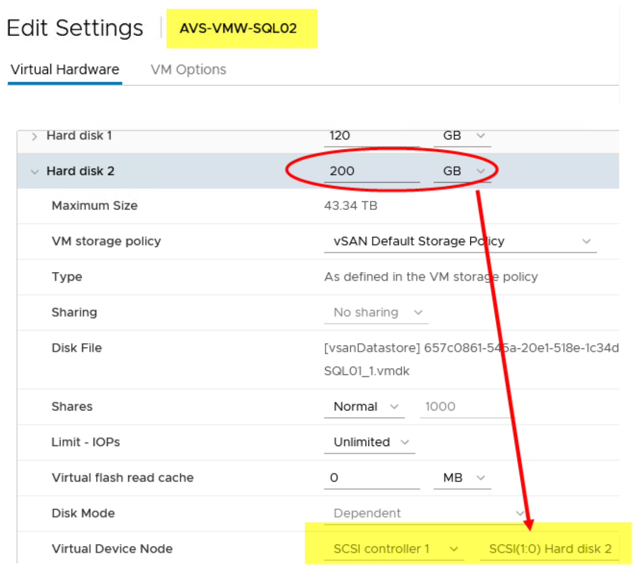
Datastores	Contents
vsanDatastore	AVS-VMW-SQL01.vmdk
.vsan.stats	AVS-VMW-SQL01_1.vmdk
AVS-VMW-CL01	
AVS-VMW-DC01	
AVS-VMW-SQL01	

• We create a new PVSCSI controller and attach the disk to it.



*New Hard disk **	200 GB
Maximum Size	43.18 TB
VM storage policy	vSAN Default Storage Policy
Sharing	Unspecified
Disk File	[vsanDatastore] 657c0861-545a-20e1-518e-1c34da421504/AVS-VMW-SQL01_1.vmdk
Shares	Normal 1000
Limit - IOPs	Unlimited
Virtual flash read cache	0 MB
Disk Mode	Dependent
Virtual Device Node	SCSI controller 0 SCSI(0:1) New Hard disk
SCSI controller 0	
*New SCSI controller **	

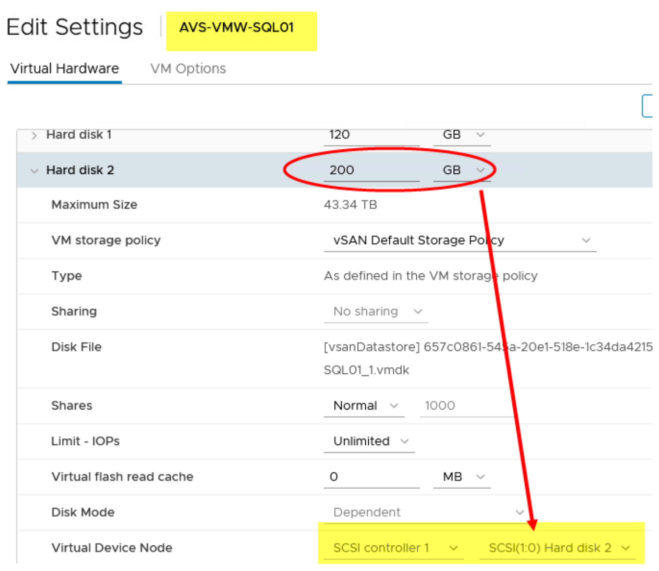
- We verify that the disk and SCSI controllers are configured identically on both VMs.



Edit Settings | AVS-VMW-SQL02

Virtual Hardware | VM Options

Hard disk 1	120 GB
Hard disk 2	200 GB
Maximum Size	43.34 TB
VM storage policy	vSAN Default Storage Policy
Type	As defined in the VM storage policy
Sharing	No sharing
Disk File	[vsanDatastore] 657c0861-545a-20e1-518e-1c34d SQL01_1.vmdk
Shares	Normal 1000
Limit - IOPs	Unlimited
Virtual flash read cache	0 MB
Disk Mode	Dependent
Virtual Device Node	SCSI controller 1 SCSI(1:0) Hard disk 2



Edit Settings | AVS-VMW-SQL01

Virtual Hardware | VM Options

Hard disk 1	120 GB
Hard disk 2	200 GB
Maximum Size	43.34 TB
VM storage policy	vSAN Default Storage Policy
Type	As defined in the VM storage policy
Sharing	No sharing
Disk File	[vsanDatastore] 657c0861-545a-20e1-518e-1c34da4215 SQL01_1.vmdk
Shares	Normal 1000
Limit - IOPs	Unlimited
Virtual flash read cache	0 MB
Disk Mode	Dependent
Virtual Device Node	SCSI controller 1 SCSI(1:0) Hard disk 2

- The End.

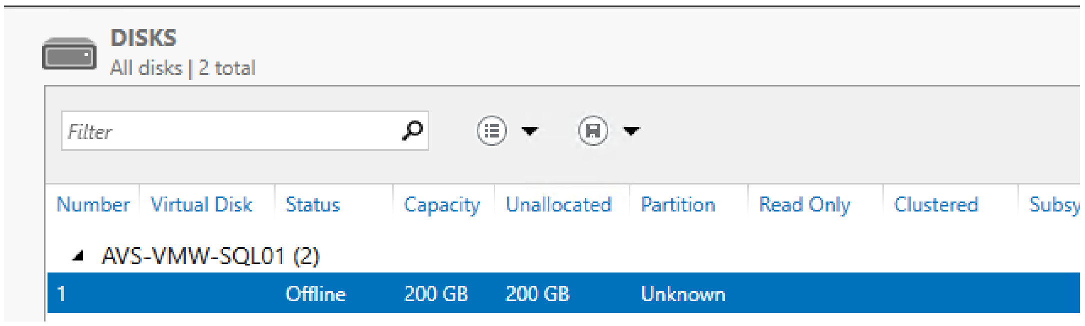
Really. That is all we need to do on the vSphere side to assign a shared disk to VMs for use in FCI configuration.

Now, we power on the VMs and let them boot into Windows.

Installing Windows Server Failover Clustering

Installing WSFC on VMs in AVS with vSphere is not in any way different from doing so on physical servers or any other platform, so we won't spend time describing it in this document. In fact, nothing in the rest of this document is peculiar to AVS or vSphere – we are including them only for guidance.

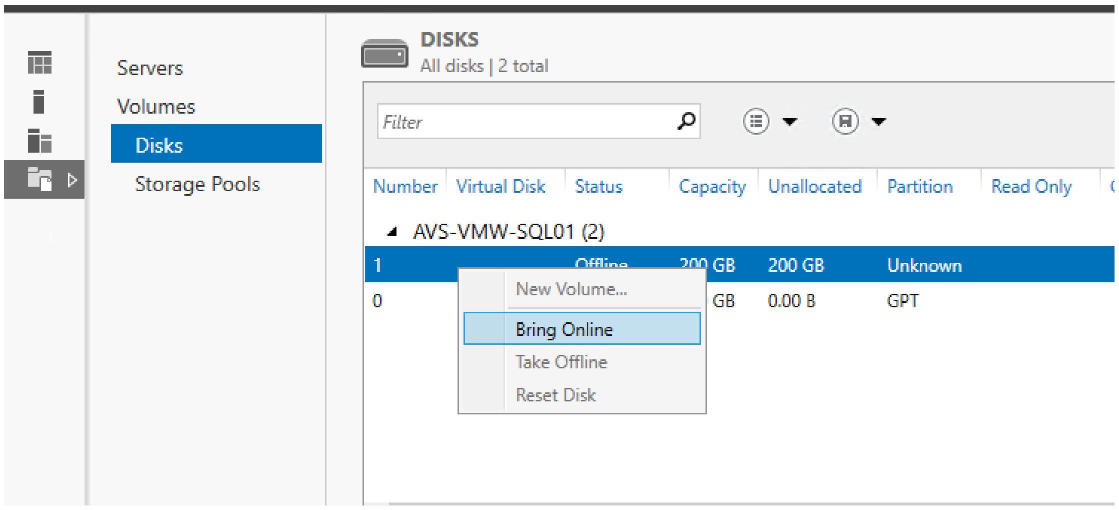
The newly-added disk is currently offline, according to Windows File and Storage Services. So, let's make it available and ready for WSFC's use (we will perform this task only on one of the nodes. There is no need to repeat the steps on any other node sharing the disk):



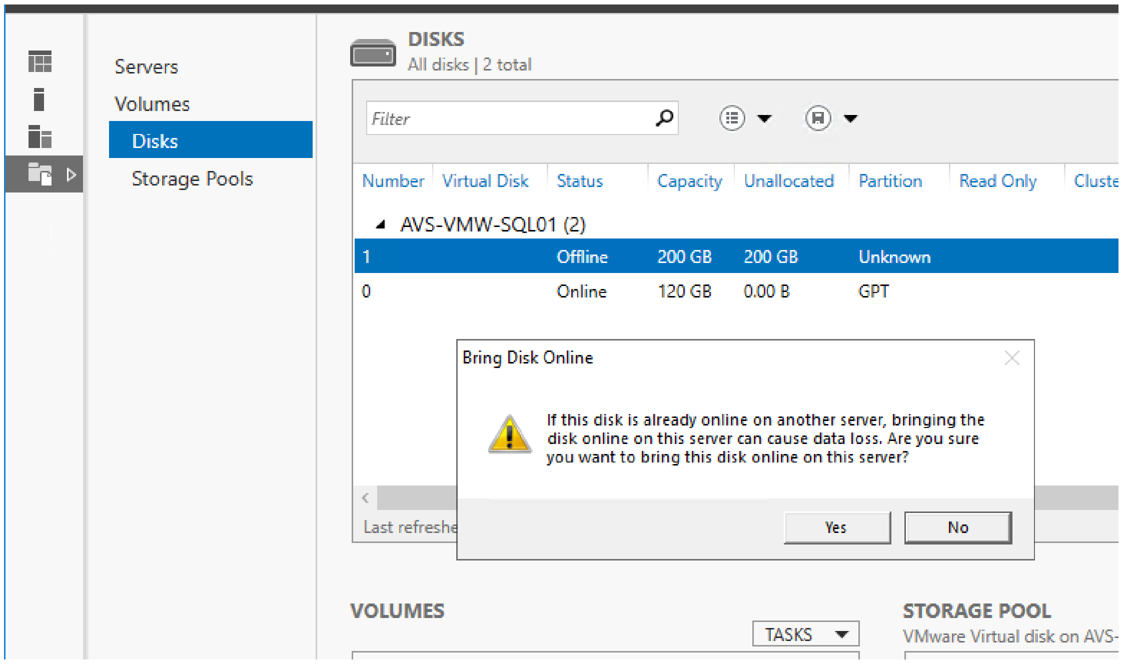
DISKS
All disks | 2 total

Number	Virtual Disk	Status	Capacity	Unallocated	Partition	Read Only	Clustered	Subsy
AVS-VMW-SQL01 (2)								
1		Offline	200 GB	200 GB	Unknown			

• We bring the disk online.



Number	Virtual Disk	Status	Capacity	Unallocated	Partition	Read Only
AVS-VMW-SQL01 (2)						
1		Offline	200 GB	200 GB	Unknown	
0			GB	0.00 B	GPT	

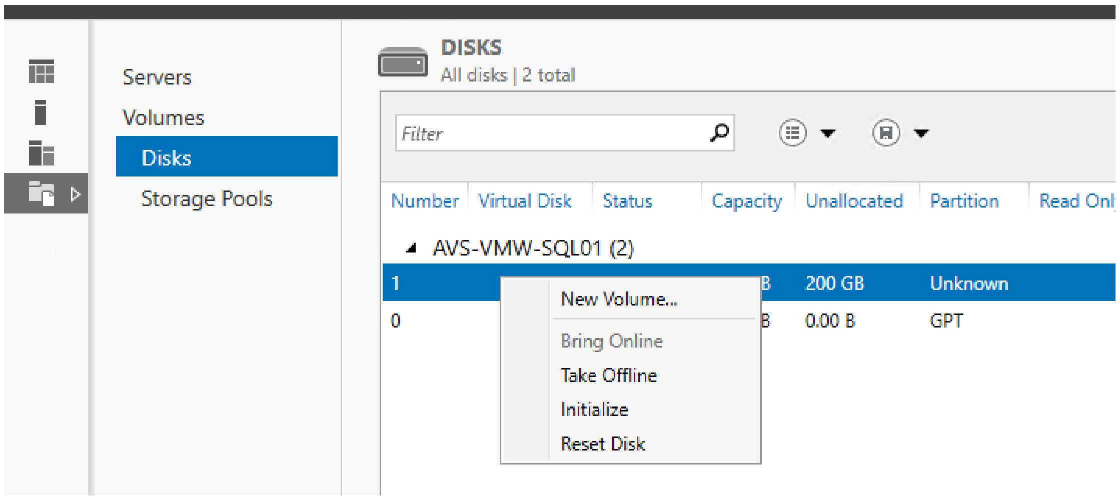


Number	Virtual Disk	Status	Capacity	Unallocated	Partition	Read Only	Cluste
AVS-VMW-SQL01 (2)							
1		Offline	200 GB	200 GB	Unknown		
0		Online	120 GB	0.00 B	GPT		

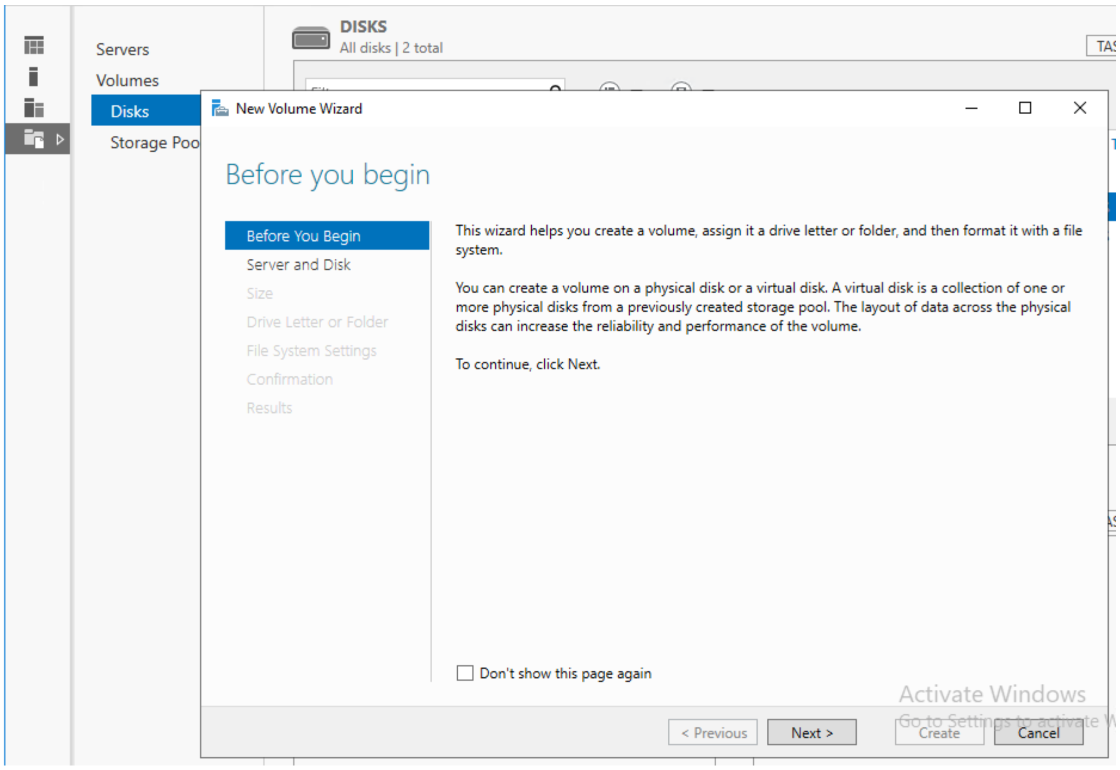
Bring Disk Online

If this disk is already online on another server, bringing the disk online on this server can cause data loss. Are you sure you want to bring this disk online on this server?

- Yes, we know what we are doing. Right?
- Now, we create a volume on it (standard Windows stuff, please feel free to skip ahead).



Number	Virtual Disk	Status	Capacity	Unallocated	Partition	Read Onl
AVS-VMW-SQL01 (2)						
1			B	200 GB	Unknown	
0			B	0.00 B	GPT	



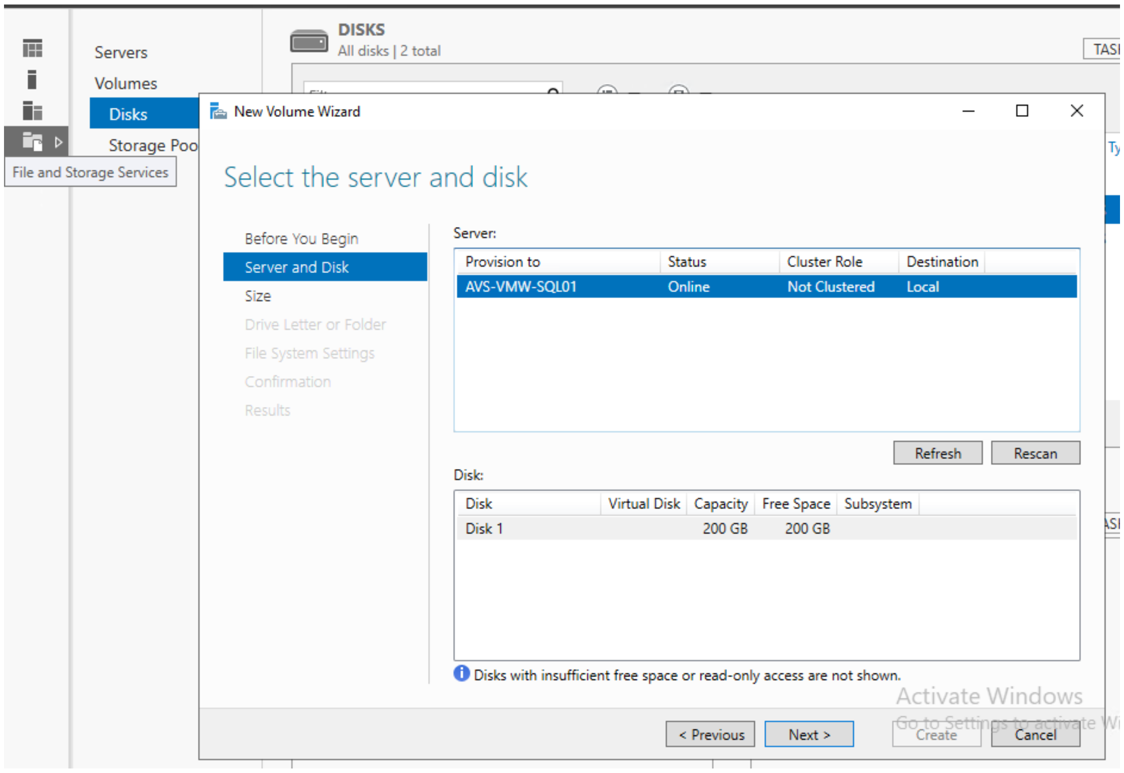
Before you begin

This wizard helps you create a volume, assign it a drive letter or folder, and then format it with a file system.

You can create a volume on a physical disk or a virtual disk. A virtual disk is a collection of one or more physical disks from a previously created storage pool. The layout of data across the physical disks can increase the reliability and performance of the volume.

To continue, click Next.

- Next



Select the server and disk

Server:

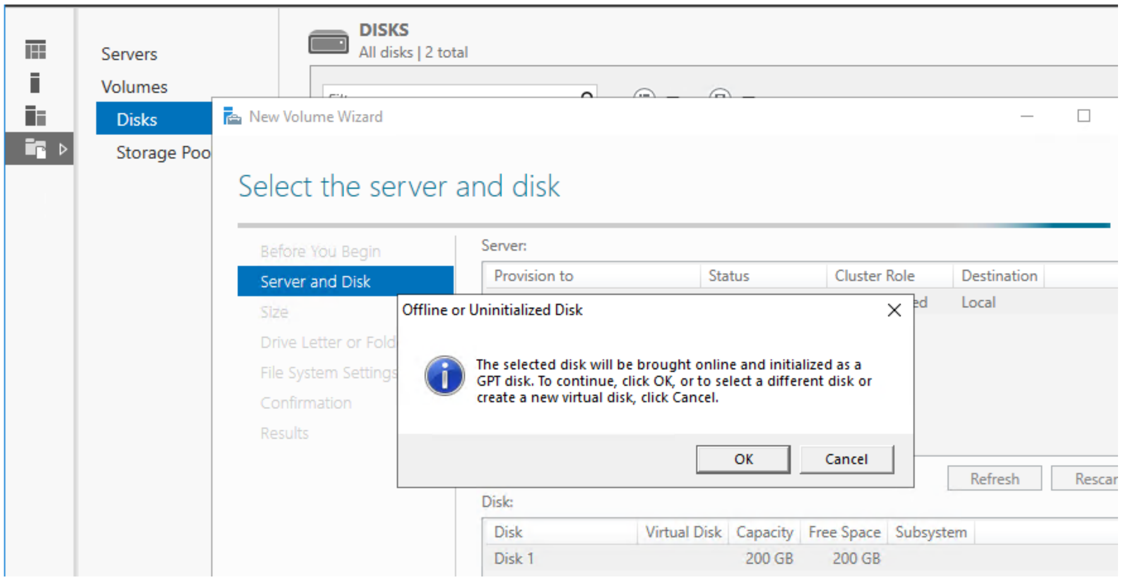
Provision to	Status	Cluster Role	Destination
AVS-VMW-SQL01	Online	Not Clustered	Local

Disk:

Disk	Virtual Disk	Capacity	Free Space	Subsystem
Disk 1		200 GB	200 GB	

Disks with insufficient free space or read-only access are not shown.

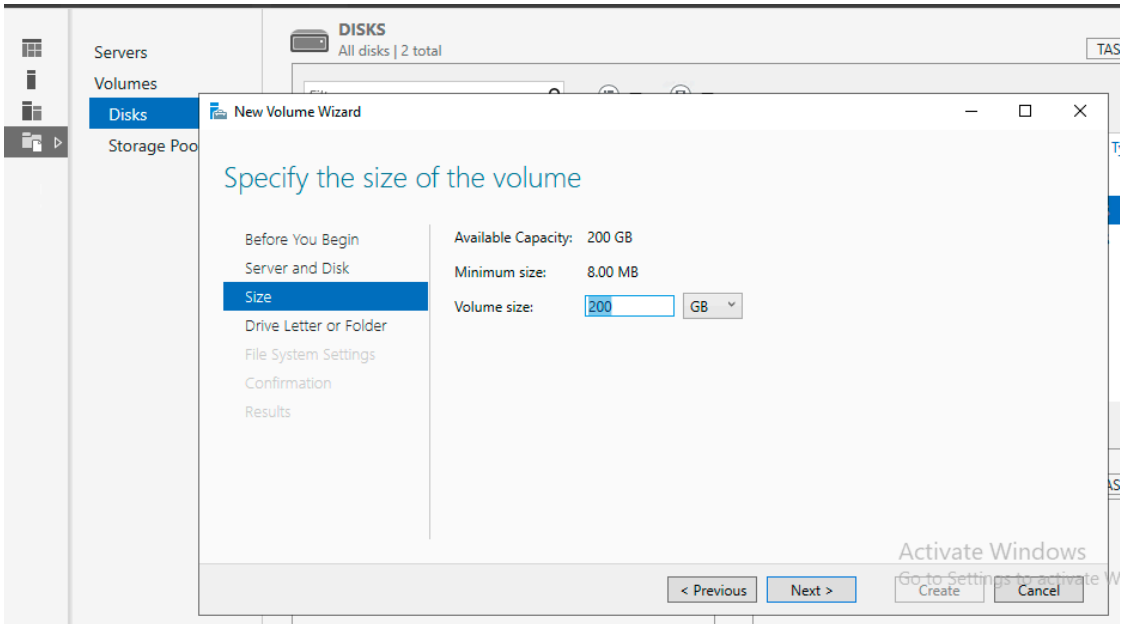
• Next



Offline or Uninitialized Disk

The selected disk will be brought online and initialized as a GPT disk. To continue, click OK, or to select a different disk or create a new virtual disk, click Cancel.

• OK



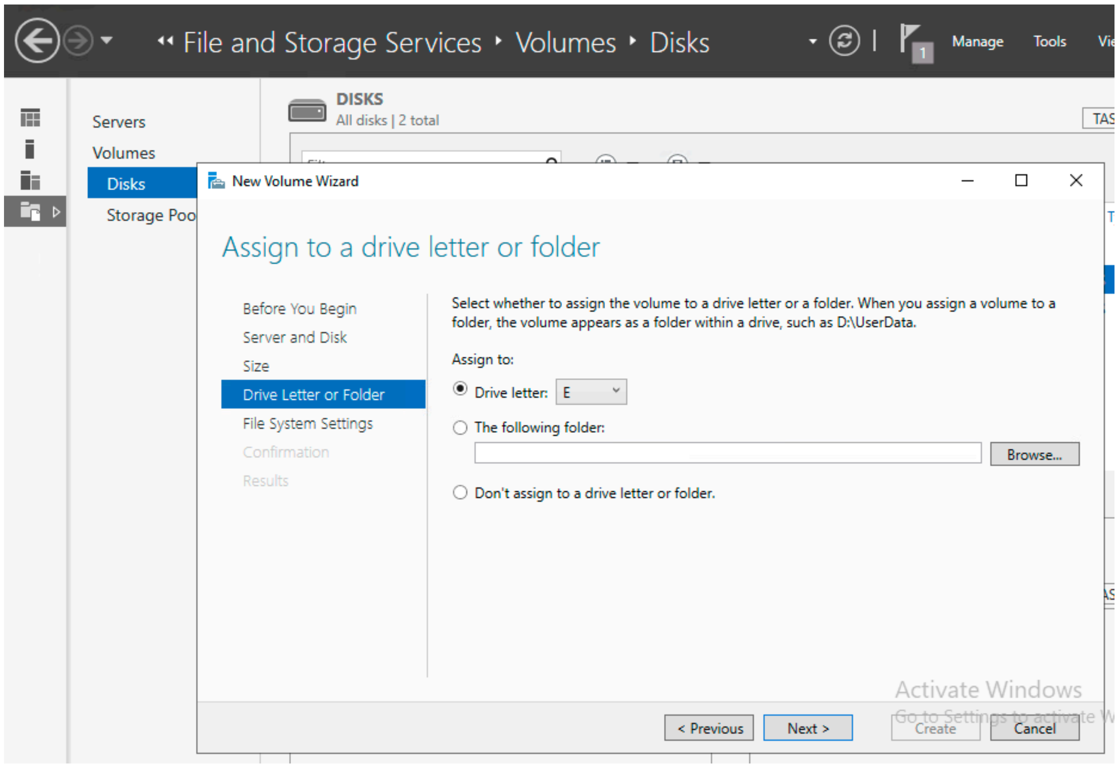
Specify the size of the volume

Available Capacity: 200 GB

Minimum size: 8.00 MB

Volume size: 200 GB

• Next



Assign to a drive letter or folder

Select whether to assign the volume to a drive letter or a folder. When you assign a volume to a folder, the volume appears as a folder within a drive, such as D:\UserData.

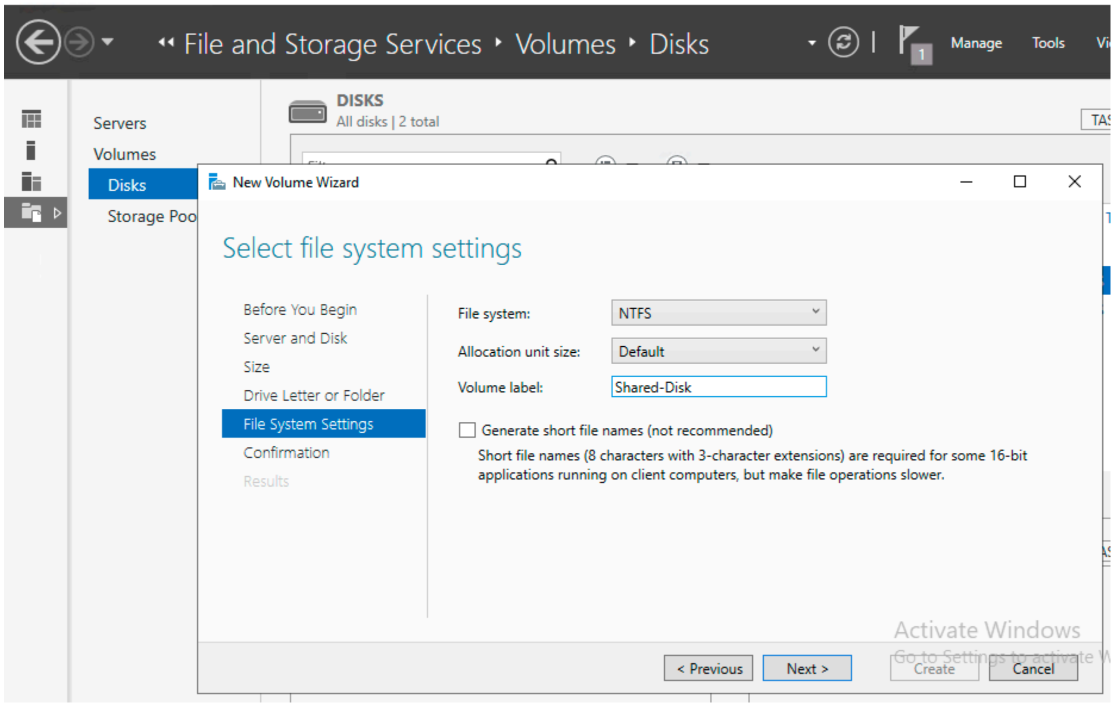
Assign to:

● Drive letter: E

○ The following folder:

○ Don't assign to a drive letter or folder.

• Next



Select file system settings

File system: NTFS

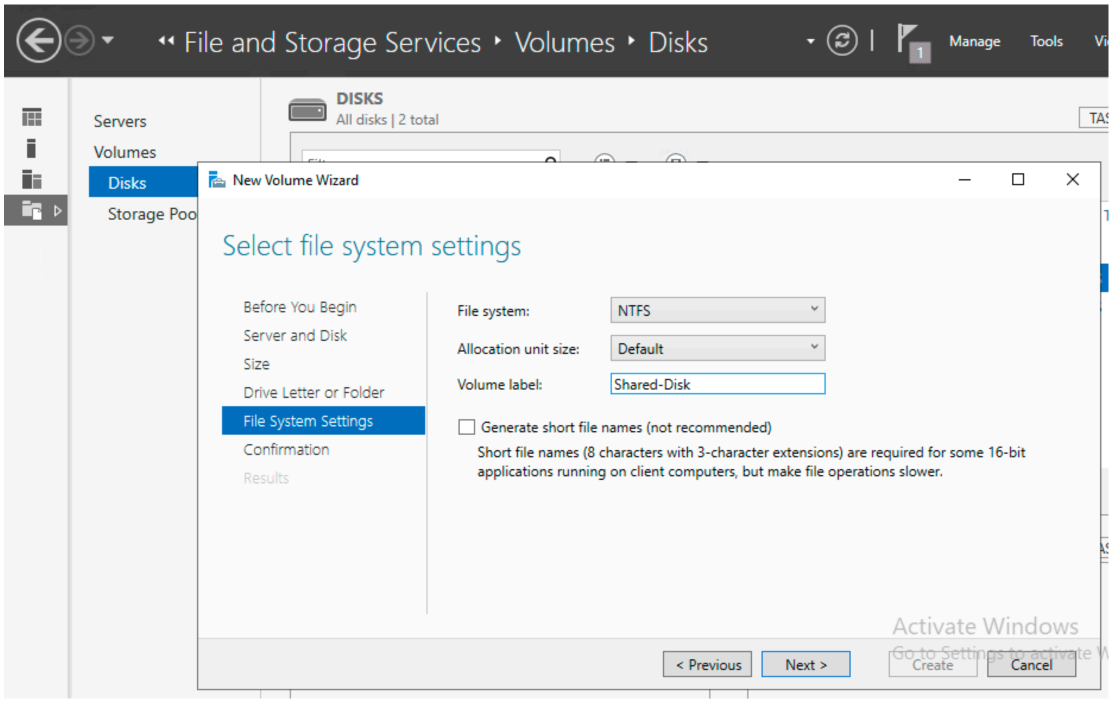
Allocation unit size: Default

Volume label: Shared-Disk

☐ Generate short file names (not recommended)

Short file names (8 characters with 3-character extensions) are required for some 16-bit applications running on client computers, but make file operations slower.

• Next



Select file system settings

File system: NTFS

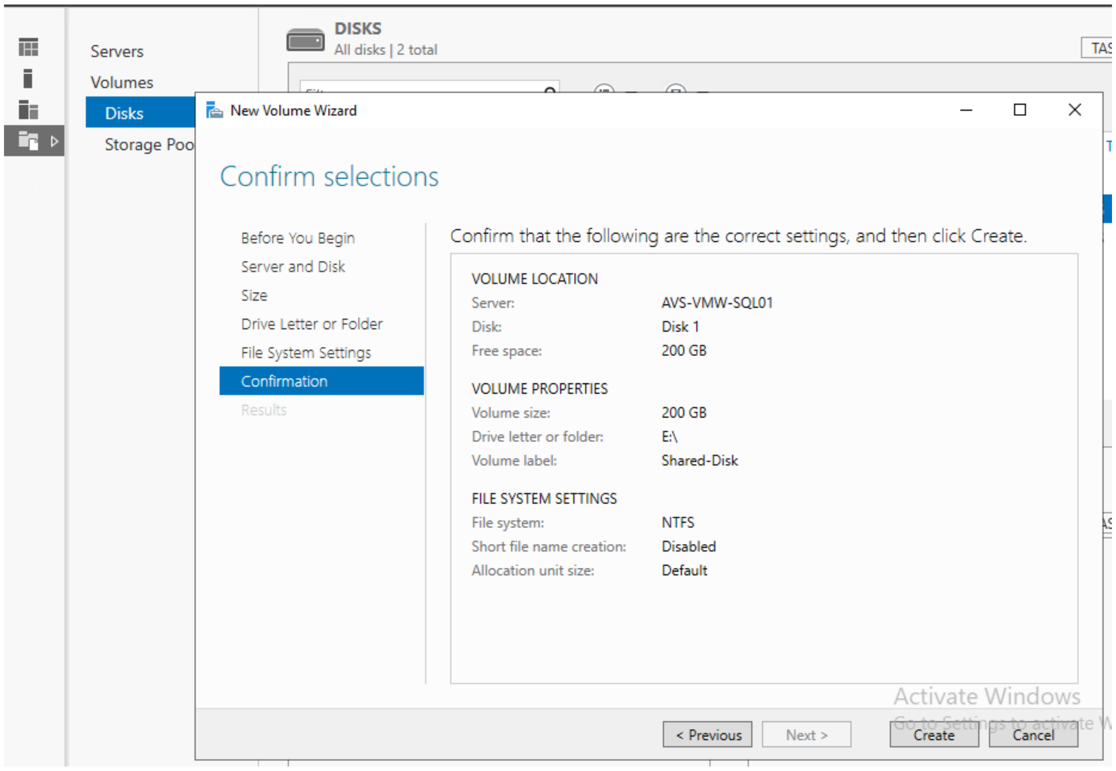
Allocation unit size: Default

Volume label: Shared-Disk

☐ Generate short file names (not recommended)

Short file names (8 characters with 3-character extensions) are required for some 16-bit applications running on client computers, but make file operations slower.

• Let's give it a fancy name and click **Next**.
• We're almost done now.



Confirm selections

Confirm that the following are the correct settings, and then click Create.

VOLUME LOCATION

Server:	AVS-VMW-SQL01
Disk:	Disk 1
Free space:	200 GB

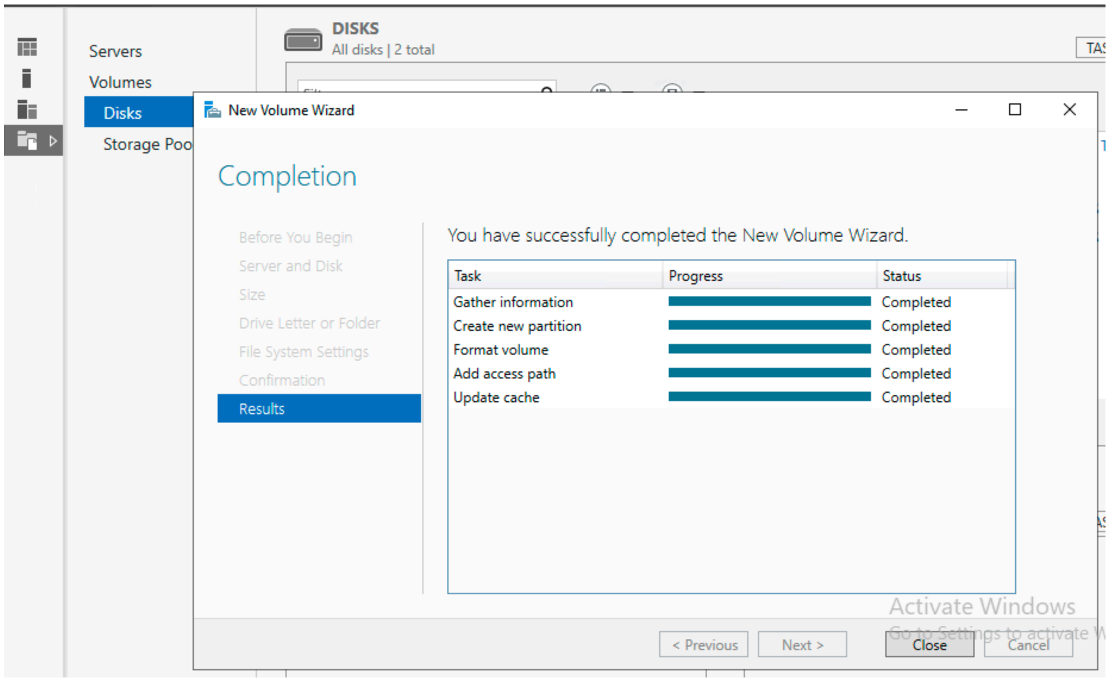
VOLUME PROPERTIES

Volume size:	200 GB
Drive letter or folder:	E:\
Volume label:	Shared-Disk

FILE SYSTEM SETTINGS

File system:	NTFS
Short file name creation:	Disabled
Allocation unit size:	Default

• Just hit **Create**, then **Next**

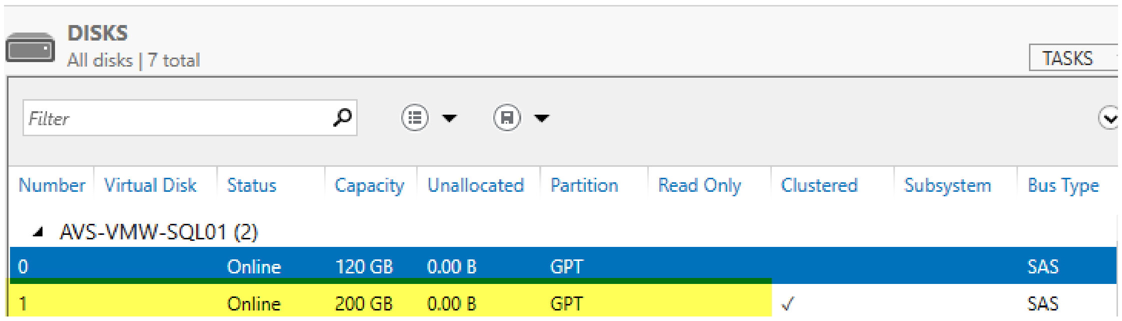


Completion

You have successfully completed the New Volume Wizard.

Task	Progress	Status
Gather information		Completed
Create new partition		Completed
Format volume		Completed
Add access path		Completed
Update cache		Completed

• Hit that **Close** button and let's get this over with.


DISKS
All disks | 7 total

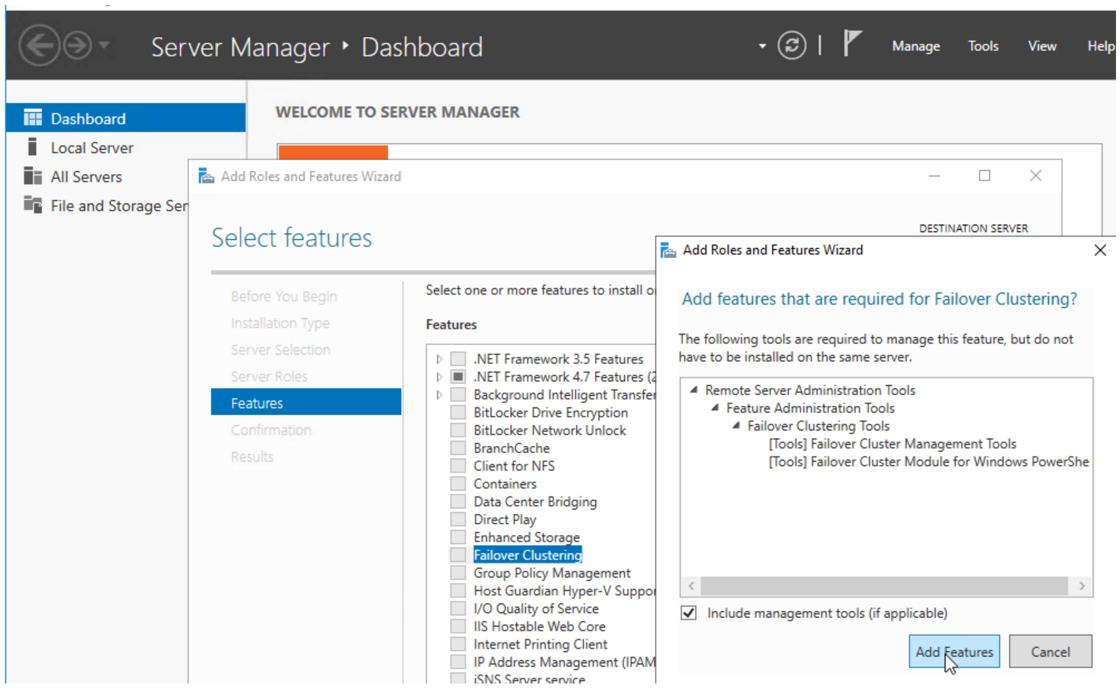
Number	Virtual Disk	Status	Capacity	Unallocated	Partition	Read Only	Clustered	Subsystem	Bus Type
AVS-VMW-SQL01 (2)									
0		Online	120 GB	0.00 B	GPT				SAS
1		Online	200 GB	0.00 B	GPT		✓		SAS

- There. Our shared disk is now ready for WSFC.

Windows Server Failover Clustering Configuration

Installing Windows Server Failover Clustering (WSFC) on a VM in AVS is not different from doing it in any other Windows Server OS Environment.

- Add WSFC as a feature in Windows.



Select features

Select one or more features to install o

Features
- .NET Framework 3.5 Features
- .NET Framework 4.7 Features (2
- Background Intelligent Transfer
- BitLocker Drive Encryption
- BitLocker Network Unlock
- BranchCache
- Client for NFS
- Containers
- Data Center Bridging
- Direct Play
- Enhanced Storage
- Failover Clustering
- Group Policy Management
- Host Guardian Hyper-V Suppor
- I/O Quality of Service
- IIS Hostable Web Core
- Internet Printing Client
- IP Address Management (IPAM
- iSNS Server service

Add Roles and Features Wizard

Add features that are required for Failover Clustering?

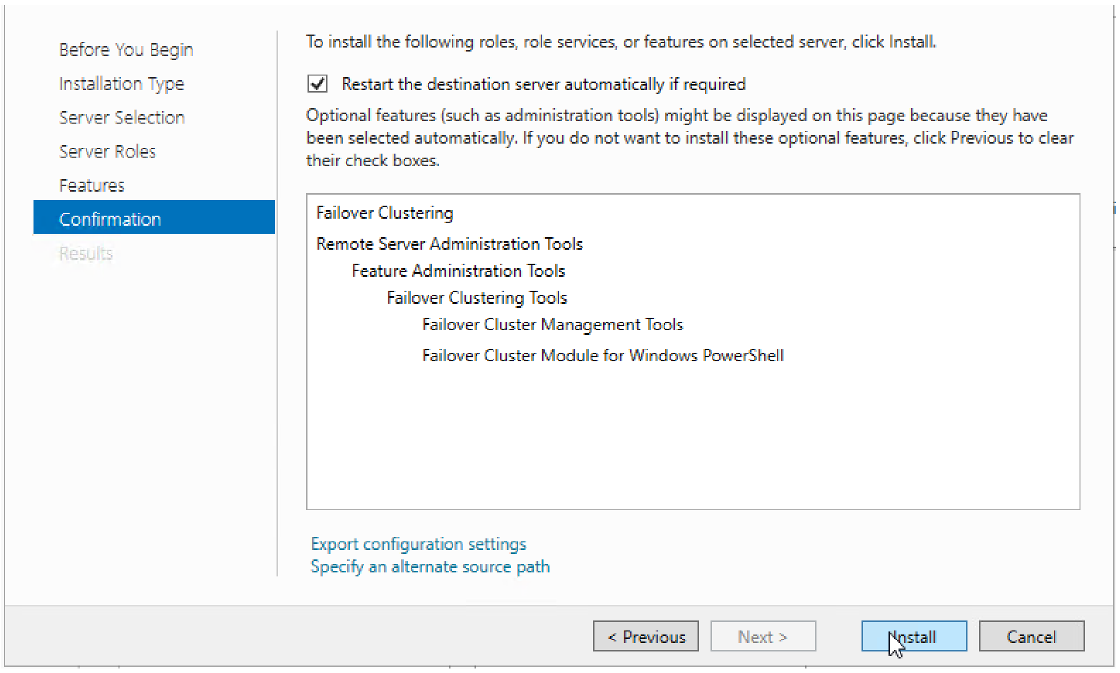
The following tools are required to manage this feature, but do not have to be installed on the same server.

- Remote Server Administration Tools
  - Feature Administration Tools
    - Failover Clustering Tools
      - [Tools] Failover Cluster Management Tools
      - [Tools] Failover Cluster Module for Windows PowerShe

☑ Include management tools (if applicable)

Add Features | Cancel

• Let's click through the process, remembering to check the option to auto-reboot upon completion.

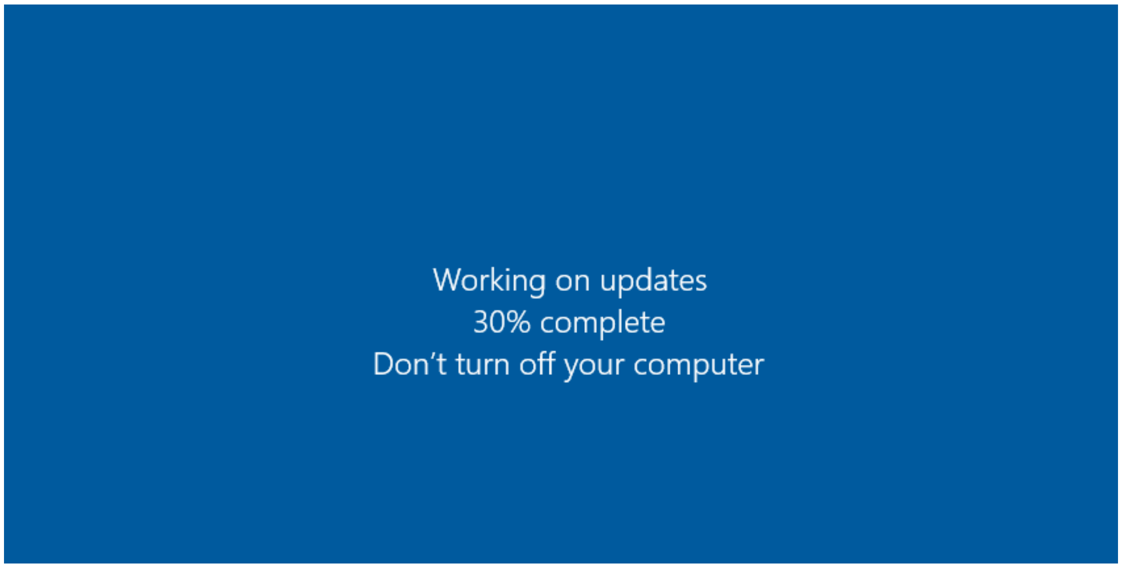


To install the following roles, role services, or features on selected server, click Install.

[✓] Restart the destination server automatically if required

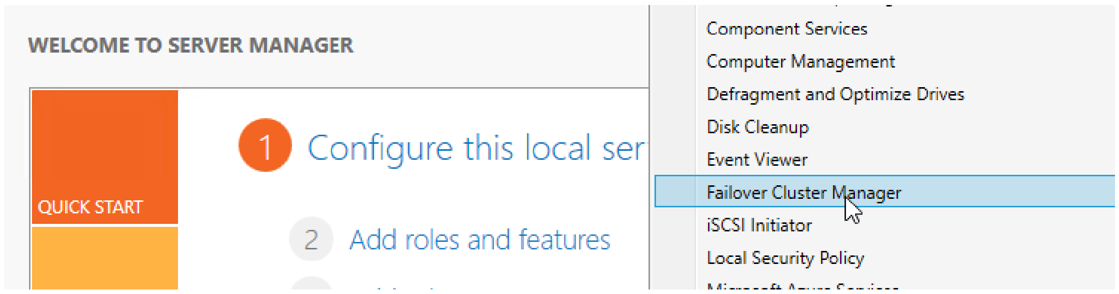
Optional features (such as administration tools) might be displayed on this page because they have been selected automatically. If you do not want to install these optional features, click Previous to clear their check boxes.

Failover Clustering
Remote Server Administration Tools
    Feature Administration Tools
        Failover Clustering Tools
            Failover Cluster Management Tools
            Failover Cluster Module for Windows PowerShell

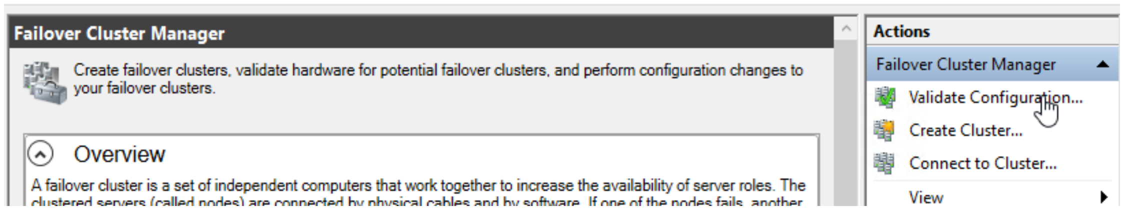


Working on updates
30% complete
Don't turn off your computer

Now, we have successfully installed WSFC. It's time to get to work.



- We begin by launching the **Cluster Validation Wizard**.

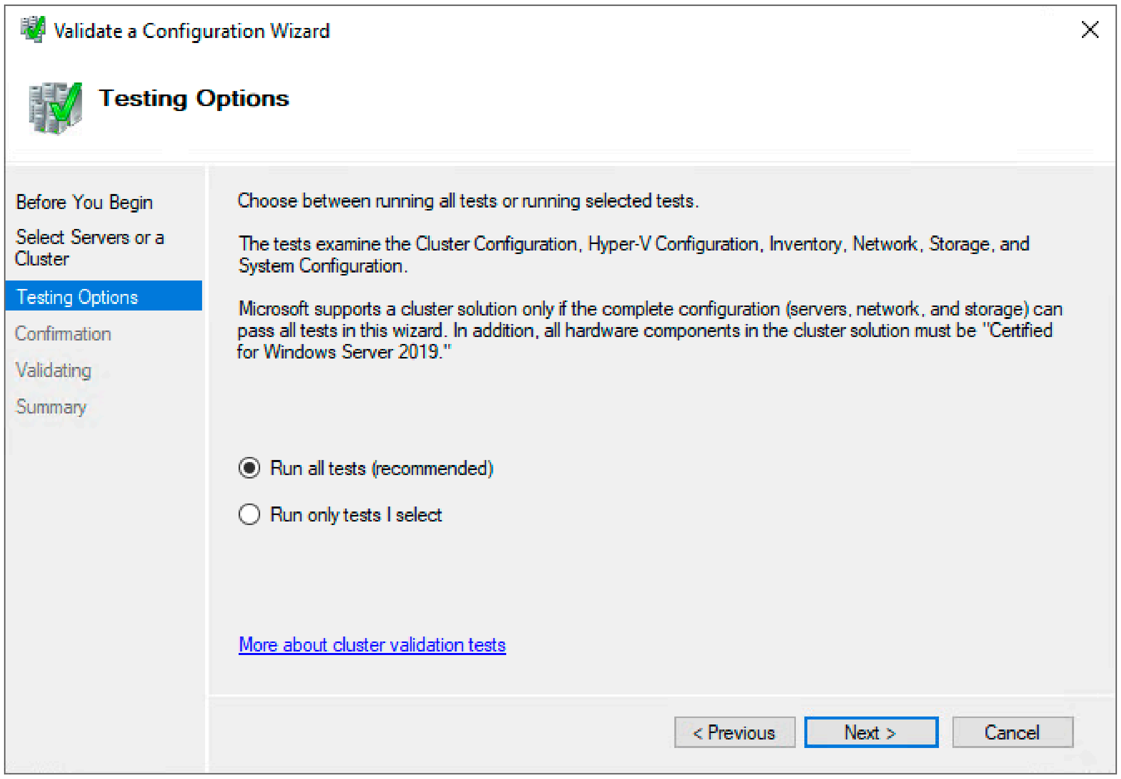


Create failover clusters, validate hardware for potential failover clusters, and perform configuration changes to your failover clusters.

Overview

A failover cluster is a set of independent computers that work together to increase the availability of server roles. The clustered servers (called nodes) are connected by physical cables and by software. If one of the nodes fails, another

- We'll let the wizard run all recommended tests, just for completeness.



Choose between running all tests or running selected tests.

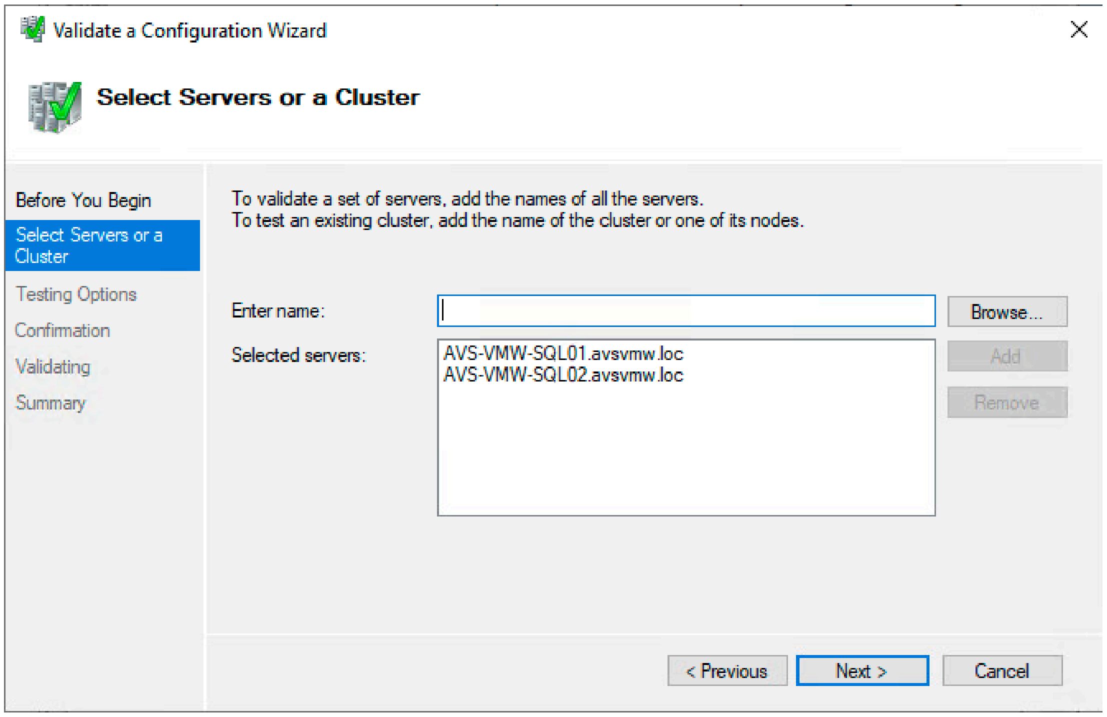
The tests examine the Cluster Configuration, Hyper-V Configuration, Inventory, Network, Storage, and System Configuration.

Microsoft supports a cluster solution only if the complete configuration (servers, network, and storage) can pass all tests in this wizard. In addition, all hardware components in the cluster solution must be "Certified for Windows Server 2019."

[✓ Run all tests (recommended)]

Run only tests I select

• We'll add the nodes we want to join into a cluster.

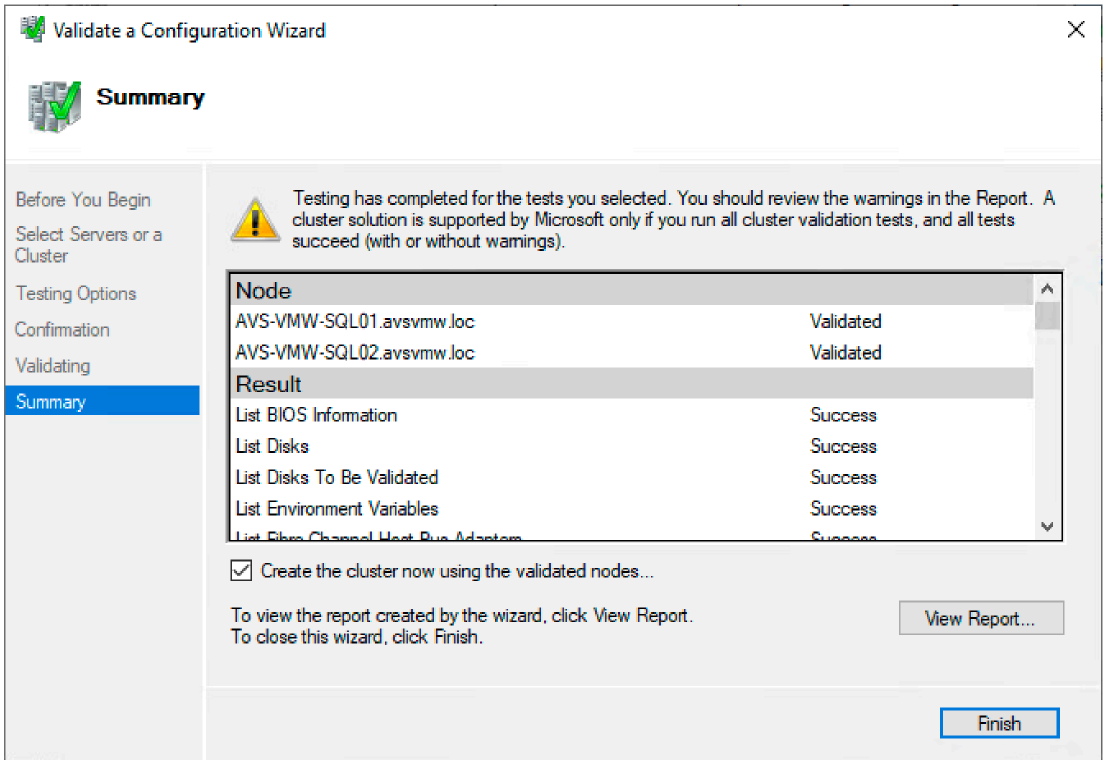


To validate a set of servers, add the names of all the servers.
To test an existing cluster, add the name of the cluster or one of its nodes.

Enter name:

Selected servers:
AVS-VMW-SQL01.avsvmw.loc
AVS-VMW-SQL02.avsvmw.loc

• Let's check the **Create the cluster now** box but, before we hit **Finish**, we'll take a look at the result of the tests by clicking the **View Report** button.



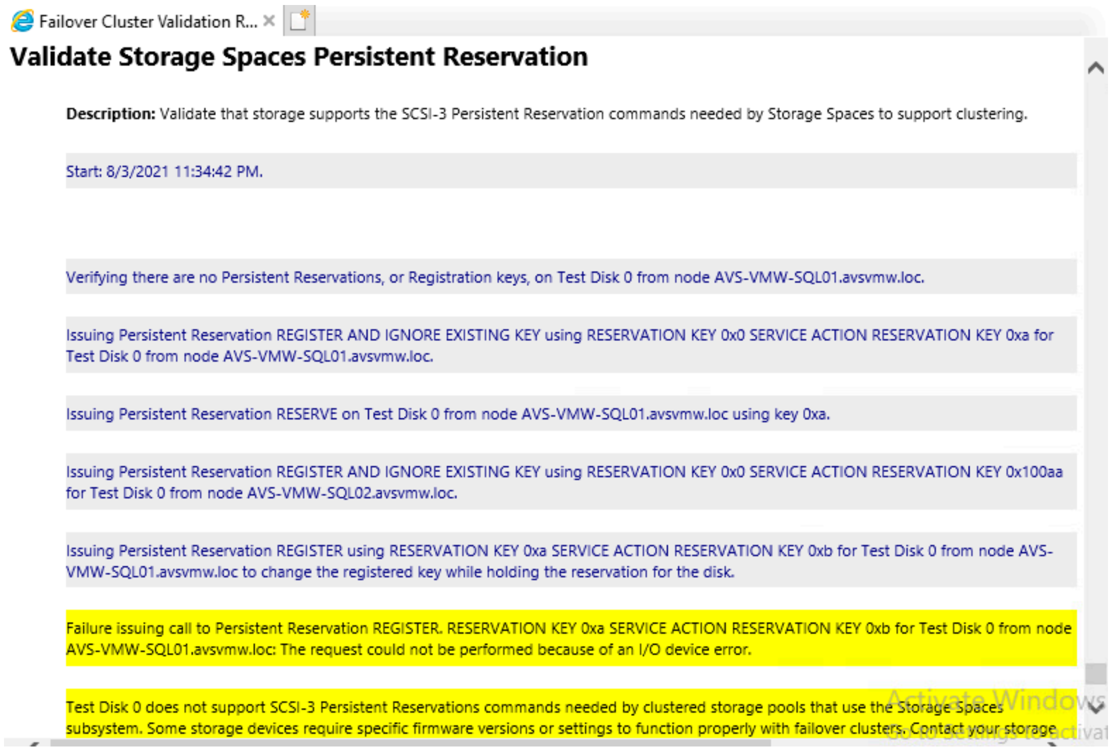
Testing has completed for the tests you selected. You should review the warnings in the Report. A cluster solution is supported by Microsoft only if you run all cluster validation tests, and all tests succeed (with or without warnings).

Node	
AVS-VMW-SQL01.avsvmw.loc	Validated
AVS-VMW-SQL02.avsvmw.loc	Validated
Result	
List BIOS Information	Success
List Disks	Success
List Disks To Be Validated	Success
List Environment Variables	Success
List Fibre Channel Host Bus Adapters	Success

☑ Create the cluster now using the validated nodes...

To view the report created by the wizard, click View Report.
To close this wizard, click Finish.

- Look at that! Some WARNINGS!!! Yes, we can safely ignore these warnings because they're from a check specific only to *Microsoft's Storage Spaces subsystems and is, therefore, irrelevant in our AVS environment*.



Failover Cluster Validation R...

Validate Storage Spaces Persistent Reservation

Description: Validate that storage supports the SCSI-3 Persistent Reservation commands needed by Storage Spaces to support clustering.

Start: 8/3/2021 11:34:42 PM.

Verifying there are no Persistent Reservations, or Registration keys, on Test Disk 0 from node AVS-VMW-SQL01.avsvmw.loc.

Issuing Persistent Reservation REGISTER AND IGNORE EXISTING KEY using RESERVATION KEY 0x0 SERVICE ACTION RESERVATION KEY 0xa for Test Disk 0 from node AVS-VMW-SQL01.avsvmw.loc.

Issuing Persistent Reservation RESERVE on Test Disk 0 from node AVS-VMW-SQL01.avsvmw.loc using key 0xa.

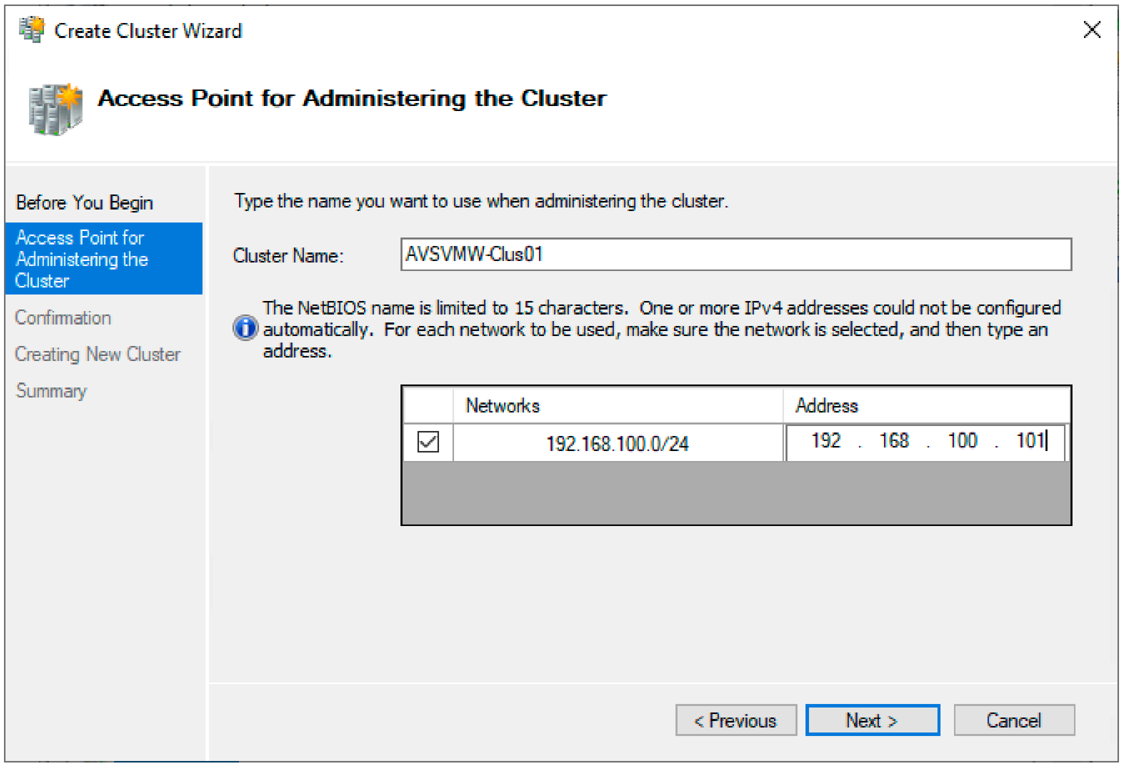
Issuing Persistent Reservation REGISTER AND IGNORE EXISTING KEY using RESERVATION KEY 0x0 SERVICE ACTION RESERVATION KEY 0x100aa for Test Disk 0 from node AVS-VMW-SQL02.avsvmw.loc.

Issuing Persistent Reservation REGISTER using RESERVATION KEY 0xa SERVICE ACTION RESERVATION KEY 0xb for Test Disk 0 from node AVS-VMW-SQL01.avsvmw.loc to change the registered key while holding the reservation for the disk.

Failure issuing call to Persistent Reservation REGISTER. RESERVATION KEY 0xa SERVICE ACTION RESERVATION KEY 0xb for Test Disk 0 from node AVS-VMW-SQL01.avsvmw.loc: The request could not be performed because of an I/O device error.

Test Disk 0 does not support SCSI-3 Persistent Reservations commands needed by clustered storage pools that use the Storage Spaces subsystem. Some storage devices require specific firmware versions or settings to function properly with failover clusters. Contact your storage

- We can close the Report and click **Finish** on the Wizard to complete our validation test and proceed to setting up the cluster.



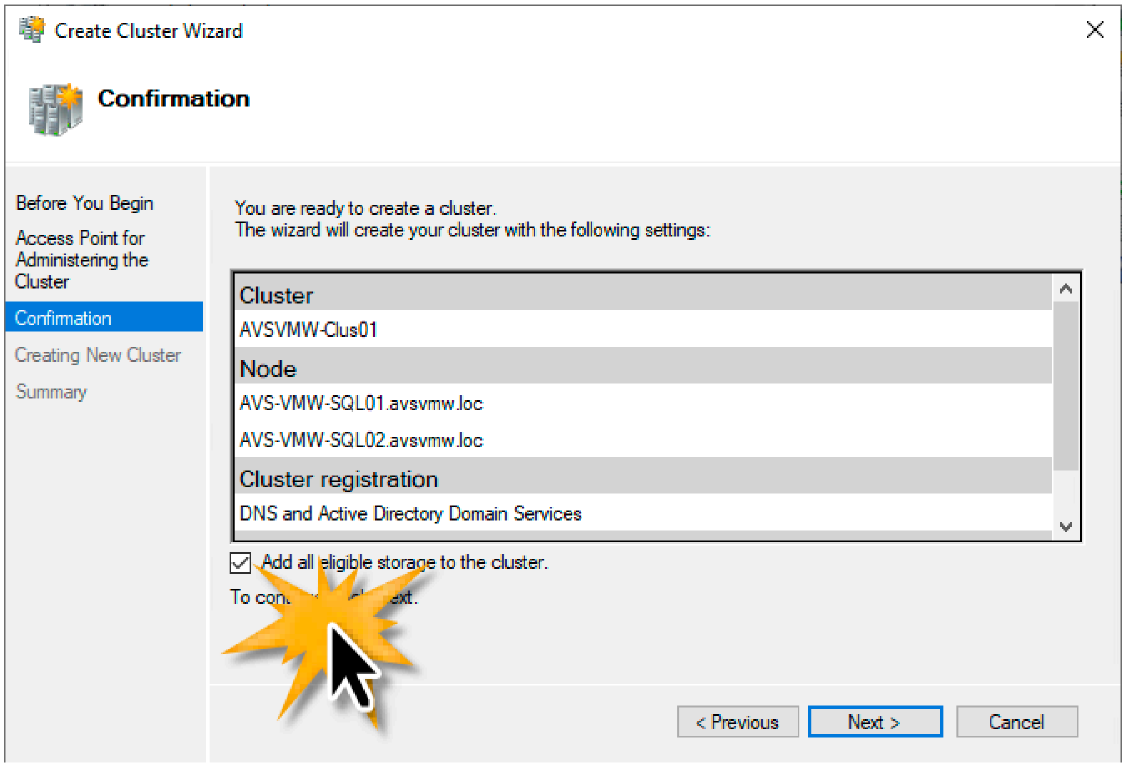
Type the name you want to use when administering the cluster.

Cluster Name: AVSVMW-Clus01

The NetBIOS name is limited to 15 characters. One or more IPv4 addresses could not be configured automatically. For each network to be used, make sure the network is selected, and then type an address.

	Networks	Address
☑	192.168.100.0/24	192 . 168 . 100 . 101

- We give the cluster's administrative access point a name and its corresponding IP.
  - **NOTE:** As mentioned previously, we assume familiarity with all of these steps, so we won't describe how to do most of the Active Directory, DNS preparatory tasks related to configuring a Windows Server Failover Cluster



Confirmation

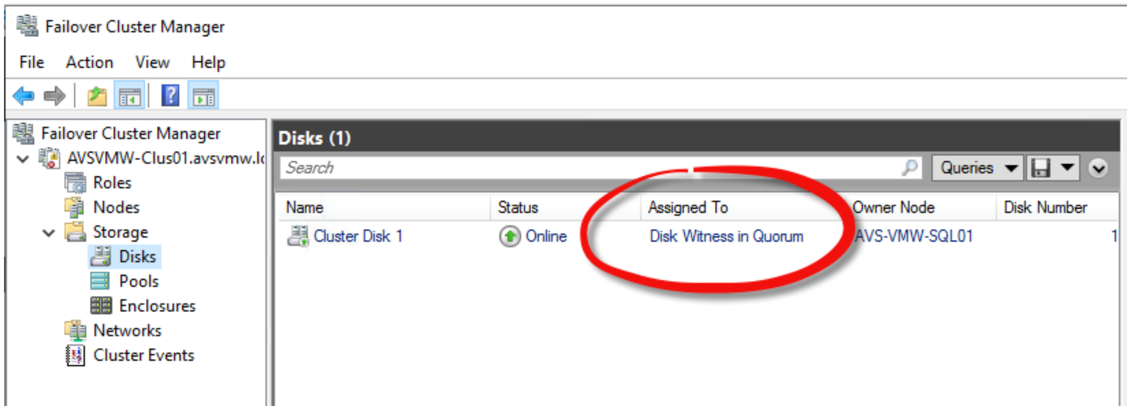
You are ready to create a cluster.
The wizard will create your cluster with the following settings:

Cluster
AVSVMW-Clus01
Node
AVS-VMW-SQL01.avsvmw.loc
AVS-VMW-SQL02.avsvmw.loc
Cluster registration
DNS and Active Directory Domain Services

☑ Add all eligible storage to the cluster.

To con... ext.

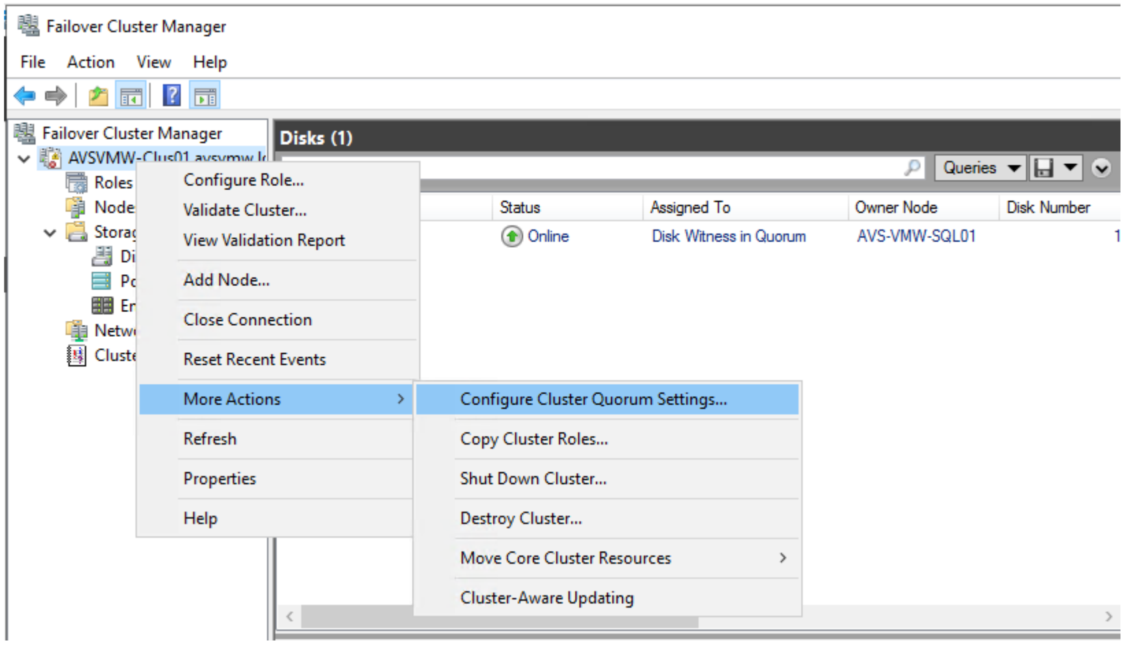
- Checking the **Add all eligible storage to the cluster** option is not recommended because, among other things, it can lead to WSFC making the wrong choice for you about the available disks and what you intend to do with them. For example, as shown in the next image, WSFC has added our Shared Disk and assigned it to the Cluster as our Witness Disk.
  - **NOTE:** Do not place the File Share Witness folder on any of the nodes participating in the Cluster. The Witness gets a vote in the Quorum, so if you put it on one of the nodes and the node becomes unavailable, you've lost two votes in the Quorum.



Disks (1)

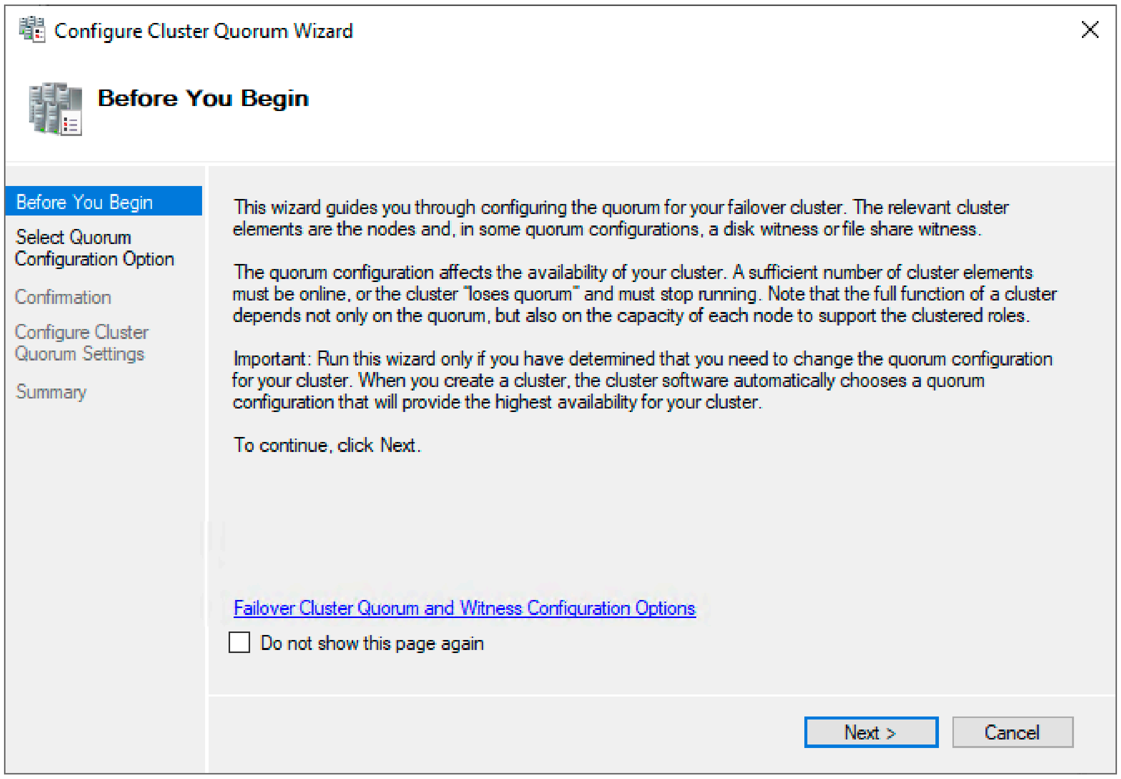
Name	Status	Assigned To	Owner Node	Disk Number
Cluster Disk 1	Online	Disk Witness in Quorum	AVS-VMW-SQL01	1

- This is definitely NOT what we want, so let's correct this quickly.
- We are going to change the Quorum setting to use a File Share Witness (FSW) instead. FSW (which is simply a folder on any server on the network, shared and made accessible to the clustered nodes and the Cluster Network Object CNO]) is easier to configure and maintain and works well for our purposes.



Status	Assigned To	Owner Node	Disk Number
Online	Disk Witness in Quorum	AVS-VMW-SQL01	1

• This involves a series of **Next, Next** clicks…


Configure Cluster Quorum Wizard
Before You Begin
Before You Begin
Select Quorum Configuration Option
Confirmation
Configure Cluster Quorum Settings
Summary
Next >
Cancel

This wizard guides you through configuring the quorum for your failover cluster. The relevant cluster elements are the nodes and, in some quorum configurations, a disk witness or file share witness.

The quorum configuration affects the availability of your cluster. A sufficient number of cluster elements must be online, or the cluster "loses quorum" and must stop running. Note that the full function of a cluster depends not only on the quorum, but also on the capacity of each node to support the clustered roles.

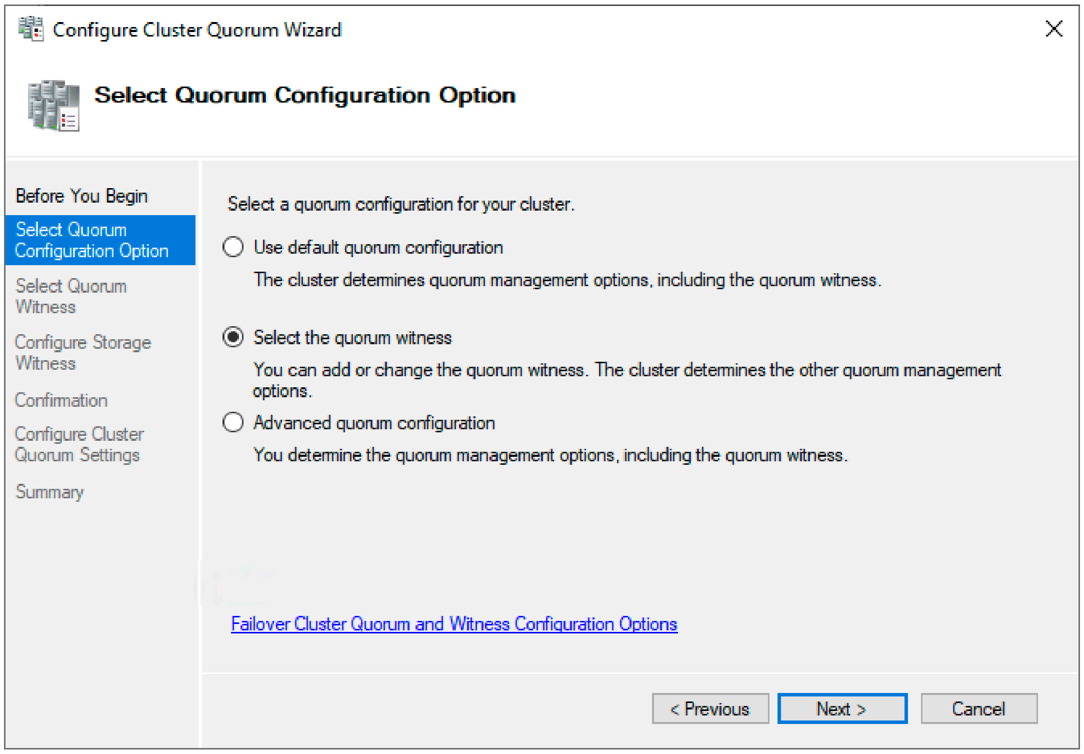
Important: Run this wizard only if you have determined that you need to change the quorum configuration for your cluster. When you create a cluster, the cluster software automatically chooses a quorum configuration that will provide the highest availability for your cluster.

To continue, click Next.

Failover Cluster Quorum and Witness Configuration Options

☐ Do not show this page again

• Click **Next**.


Configure Cluster Quorum Wizard

Select Quorum Configuration Option

Before You Begin
Select Quorum Configuration Option
Select Quorum Witness
Configure Storage Witness
Confirmation
Configure Cluster Quorum Settings
Summary

Select a quorum configuration for your cluster.

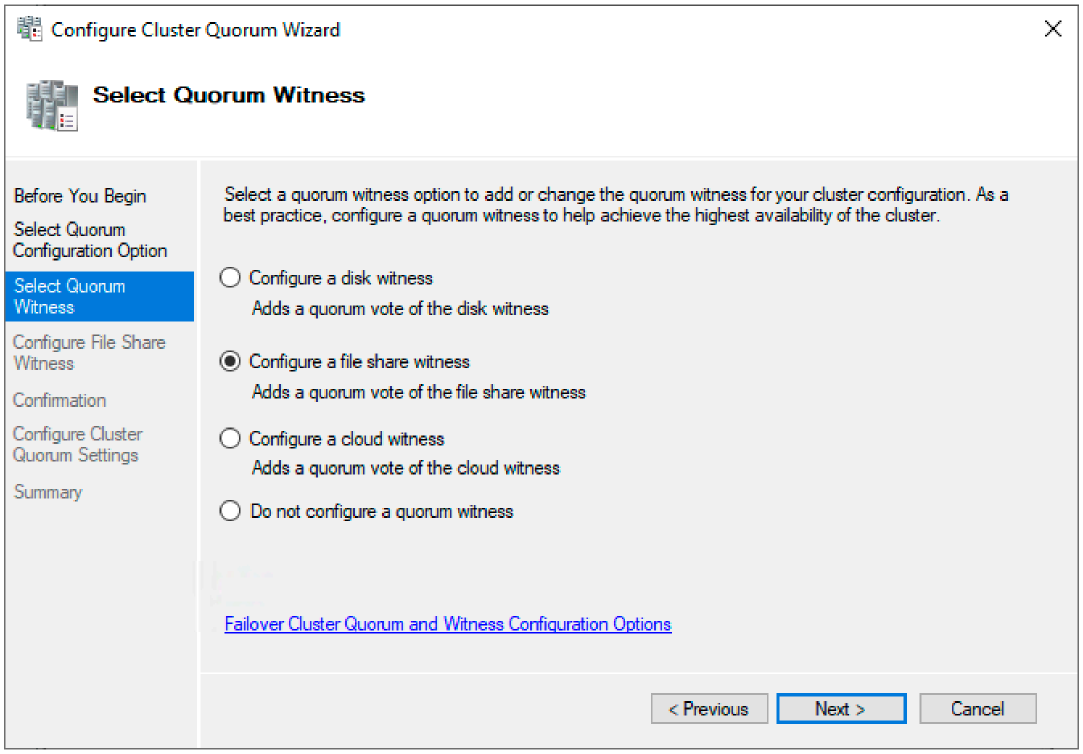
( ) Use default quorum configuration
The cluster determines quorum management options, including the quorum witness.

(●) Select the quorum witness
You can add or change the quorum witness. The cluster determines the other quorum management options.

( ) Advanced quorum configuration
You determine the quorum management options, including the quorum witness.

Failover Cluster Quorum and Witness Configuration Options

• Choose the **Select Quorum Witness** option, then click **Next**.


Configure Cluster Quorum Wizard

Select Quorum Witness

Before You Begin
Select Quorum Configuration Option
Select Quorum Witness
Configure File Share Witness
Confirmation
Configure Cluster Quorum Settings
Summary

Select a quorum witness option to add or change the quorum witness for your cluster configuration. As a best practice, configure a quorum witness to help achieve the highest availability of the cluster.

( ) Configure a disk witness
Adds a quorum vote of the disk witness

(●) Configure a file share witness
Adds a quorum vote of the file share witness

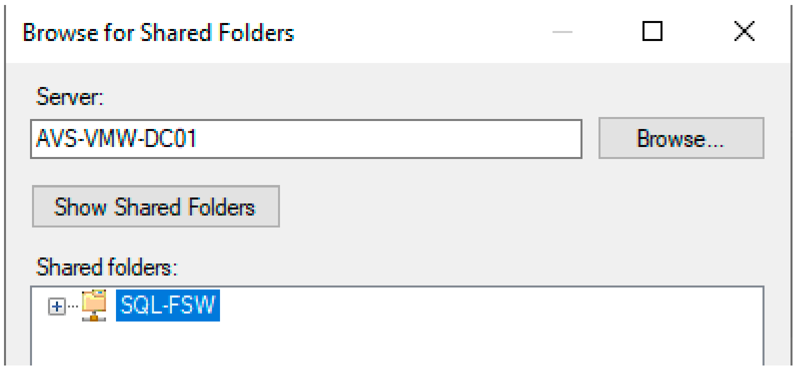
( ) Configure a cloud witness
Adds a quorum vote of the cloud witness

( ) Do not configure a quorum witness

Failover Cluster Quorum and Witness Configuration Options

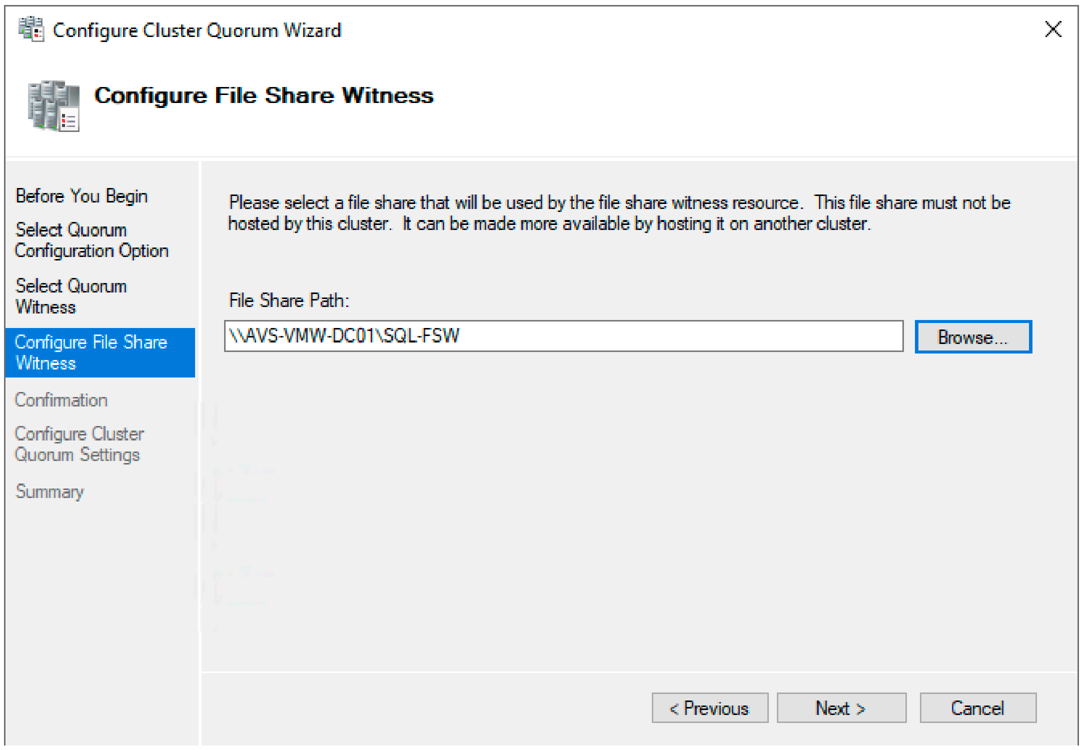
• We want the **File Share Witness** option, so check that and click **Next**.

- Browse to where the share is located, then select it.



Server: AVS-VMW-DC01

Shared folders: SQL-FSW

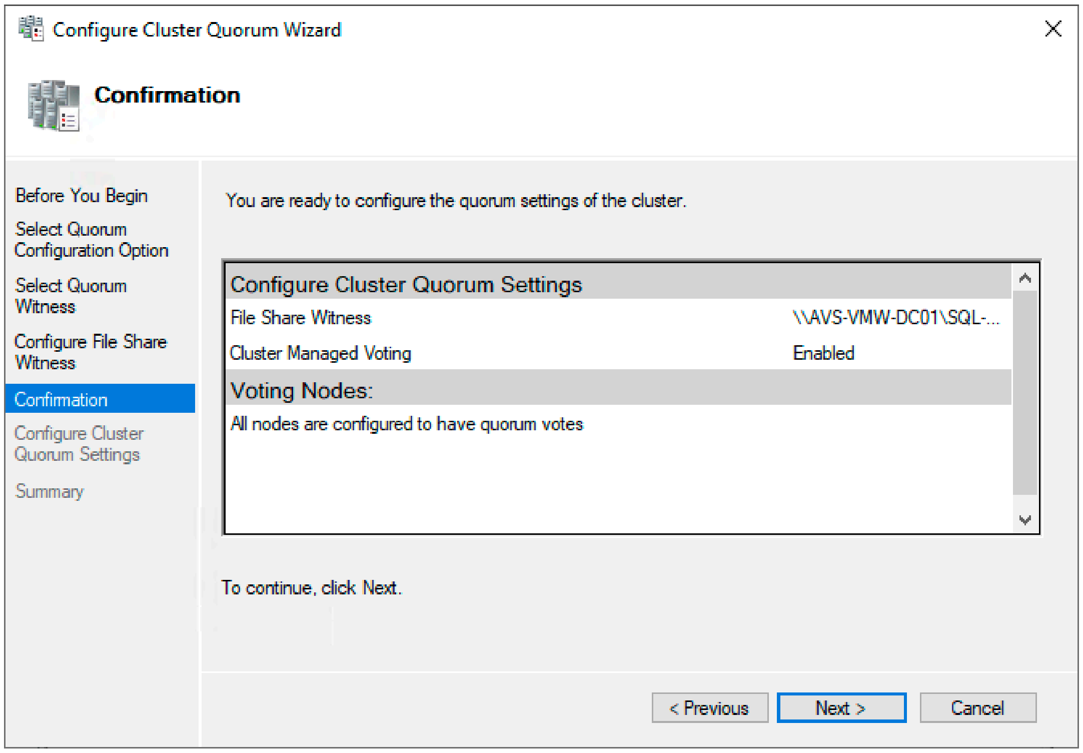

Configure Cluster Quorum Wizard
Before You Begin
Select Quorum Configuration Option
Select Quorum Witness
Configure File Share Witness
Confirmation
Configure Cluster Quorum Settings
Summary
Browse...
< Previous
Next >
Cancel

Configure File Share Witness

Please select a file share that will be used by the file share witness resource. This file share must not be hosted by this cluster. It can be made more available by hosting it on another cluster.

File Share Path: \\AVS-VMW-DC01\SQL-FSW

- We confirm that we have selected the correct share, then we click **Next**.

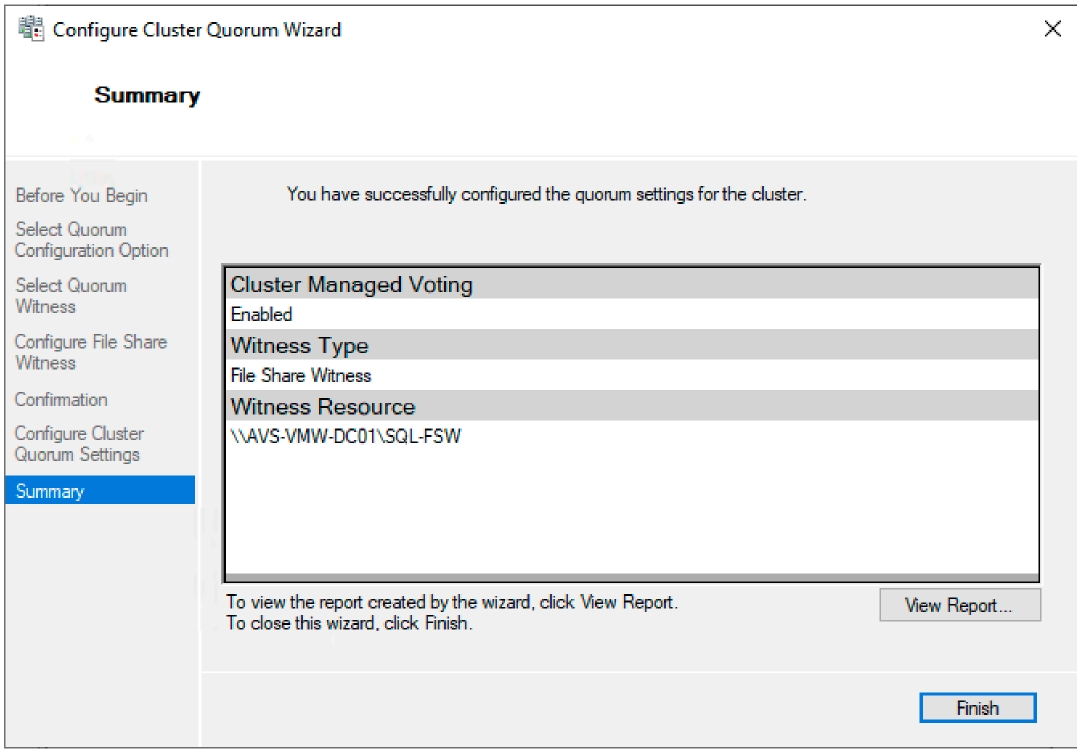

Configure Cluster Quorum Wizard
Confirmation
Before You Begin
Select Quorum Configuration Option
Select Quorum Witness
Configure File Share Witness
Confirmation
Configure Cluster Quorum Settings
Summary
< Previous
Next >
Cancel

You are ready to configure the quorum settings of the cluster.

Configure Cluster Quorum Settings	
File Share Witness	\\AVS-VMW-DC01\SQL-...
Cluster Managed Voting	Enabled
Voting Nodes:	
All nodes are configured to have quorum votes	

To continue, click Next.

• One more **Next**

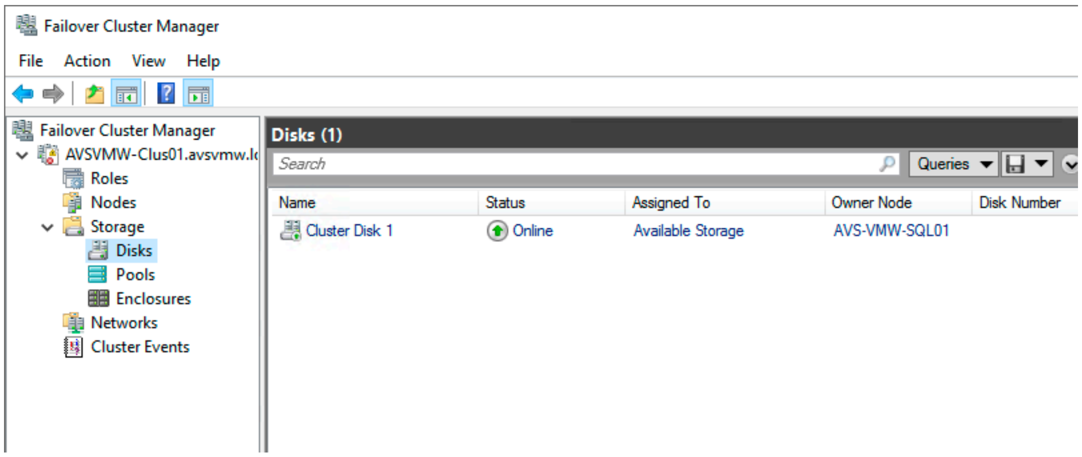

Configure Cluster Quorum Wizard
Summary
Before You Begin
Select Quorum Configuration Option
Select Quorum Witness
Configure File Share Witness
Confirmation
Configure Cluster Quorum Settings
Summary
View Report...
Finish

You have successfully configured the quorum settings for the cluster.

Cluster Managed Voting
Enabled
Witness Type
File Share Witness
Witness Resource
\\AVS-VMW-DC01\SQL-FSW

To view the report created by the wizard, click View Report.
To close this wizard, click Finish.

• And, we wrap it up with a **Finish** (unless you'd like to read the Report, of course).



Failover Cluster Manager

File Action View Help

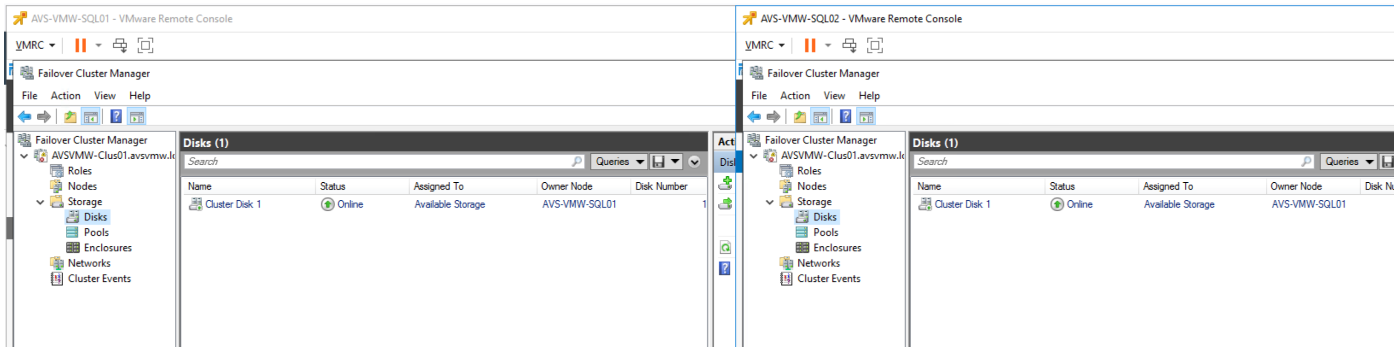
Disks (1)

Name	Status	Assigned To	Owner Node	Disk Number
Cluster Disk 1	Online	Available Storage	AVS-VMW-SQL01	

- Now, our Shared Disk is recognized by WSFC as we intended for it to be – **Available Storage**. We can now use it for our Microsoft SQL Server FCI-clustering purposes.

But, before we do that, why don't we confirm that the disk is actually a clustered resource and can survive the failure of one of the nodes we just clustered? This would give us the peace of mind we need before laying our SQL Server bits on top.

Here is how both nodes see the shared disk after we've completed the cluster setup.



AVS-VMW-SQL01 - VMware Remote Console

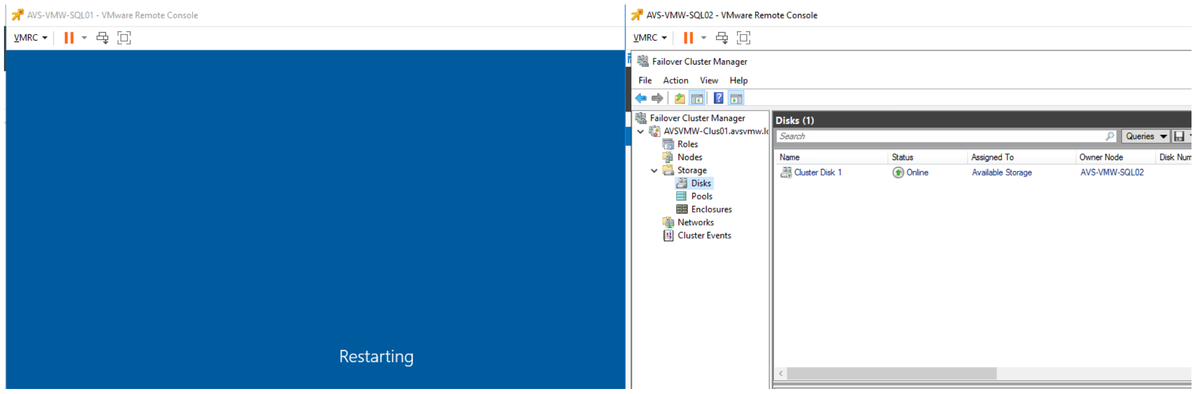
Name	Status	Assigned To	Owner Node	Disk Number
Cluster Disk 1	Online	Available Storage	AVS-VMW-SQL01	1

AVS-VMW-SQL02 - VMware Remote Console

Name	Status	Assigned To	Owner Node
Cluster Disk 1	Online	Available Storage	AVS-VMW-SQL01

- Both nodes see that SQL01 currently owns the disk, which is fine.

But what happens if SQL01 were to become suddenly unavailable?



AVS-VMW-SQL01 - VMware Remote Console

Restarting

AVS-VMW-SQL02 - VMware Remote Console

Name	Status	Assigned To	Owner Node
Cluster Disk 1	Online	Available Storage	AVS-VMW-SQL02

- WSFC does what it's supposed to do – it lets SQL02 take ownership of the clustered resource.

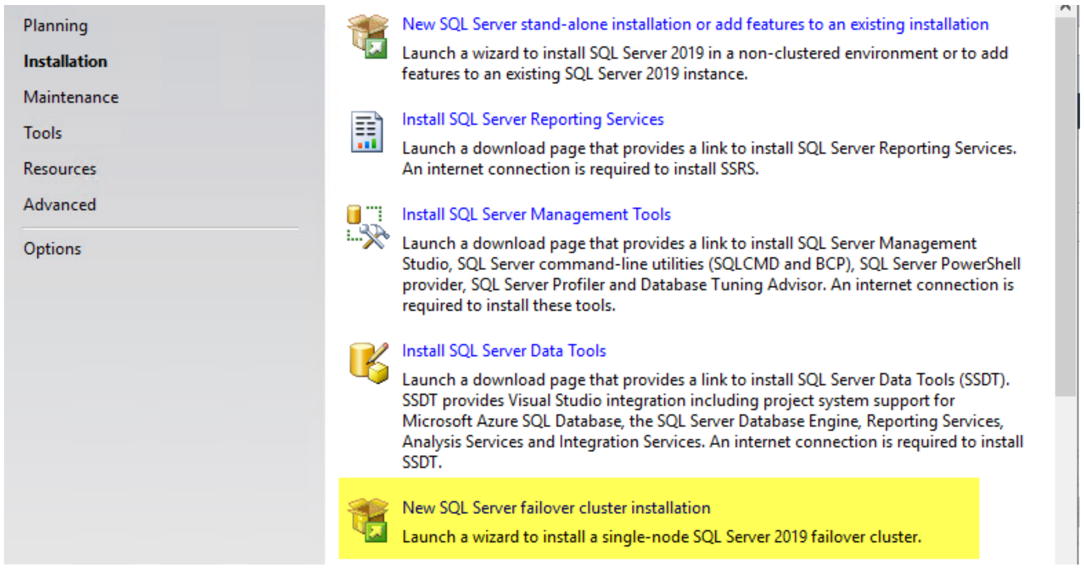
At this point, we have completed our demonstration of clustering shared resources (disk) for multiple Windows Server Failover Clustering nodes on Azure VMware Solutions. As you can see, the actual WSFC installation and configuration procedures are similar, regardless of the platform.

We are now going to simulate and document layering Microsoft SQL Server instances and databases on top of the clustered infrastructure.

– **NOTE:** As before, we assume prior knowledge of the administrative steps and tasks involved in setting up FCI and availability group in a typical Microsoft SQL Server environment, so we will not be describing these in detail.

Installing and Configuring Microsoft SQL Server Always On Failover Clustering Instance on AVS

We will install Microsoft SQL Server first on SQL01, then on SQL02. The major difference in the installation process is that we will select the **New SQL Server Failover Cluster installation** option for SQL01 and the **Add node to a SQL Server Failover Cluster** option for SQL02 as shown below:



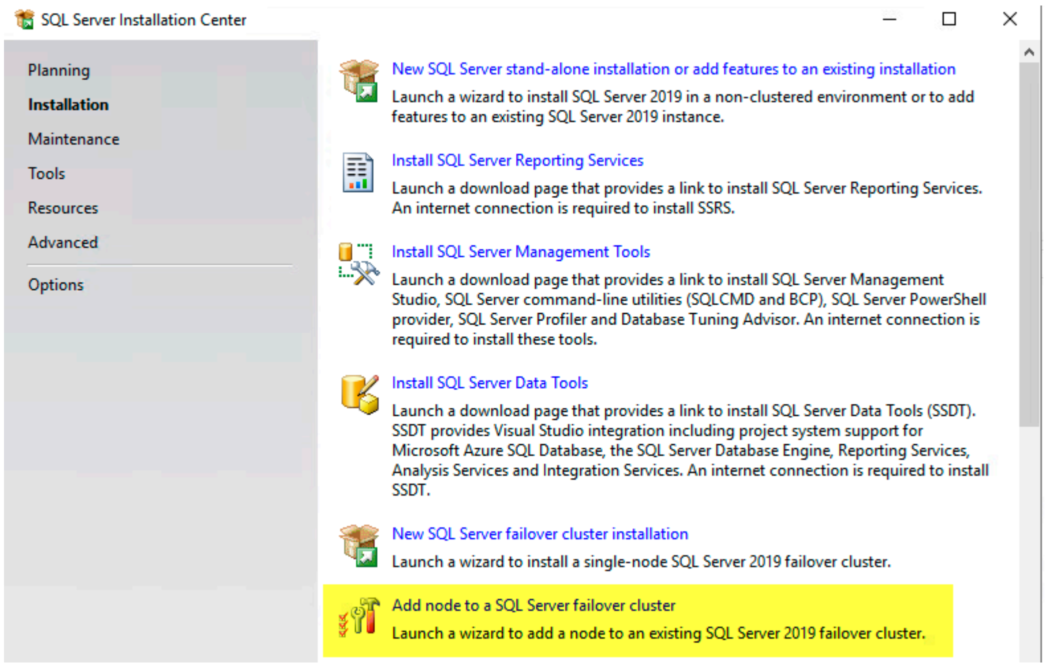
New SQL Server stand-alone installation or add features to an existing installation
Launch a wizard to install SQL Server 2019 in a non-clustered environment or to add features to an existing SQL Server 2019 instance.

Install SQL Server Reporting Services
Launch a download page that provides a link to install SQL Server Reporting Services. An internet connection is required to install SSRS.

Install SQL Server Management Tools
Launch a download page that provides a link to install SQL Server Management Studio, SQL Server command-line utilities (SQLCMD and BCP), SQL Server PowerShell provider, SQL Server Profiler and Database Tuning Advisor. An internet connection is required to install these tools.

Install SQL Server Data Tools
Launch a download page that provides a link to install SQL Server Data Tools (SSDT). SSDT provides Visual Studio integration including project system support for Microsoft Azure SQL Database, the SQL Server Database Engine, Reporting Services, Analysis Services and Integration Services. An internet connection is required to install SSDT.

New SQL Server failover cluster installation
Launch a wizard to install a single-node SQL Server 2019 failover cluster.



SQL Server Installation Center

Planning
Installation
Maintenance
Tools
Resources
Advanced
Options

New SQL Server stand-alone installation or add features to an existing installation
Launch a wizard to install SQL Server 2019 in a non-clustered environment or to add features to an existing SQL Server 2019 instance.

Install SQL Server Reporting Services
Launch a download page that provides a link to install SQL Server Reporting Services. An internet connection is required to install SSRS.

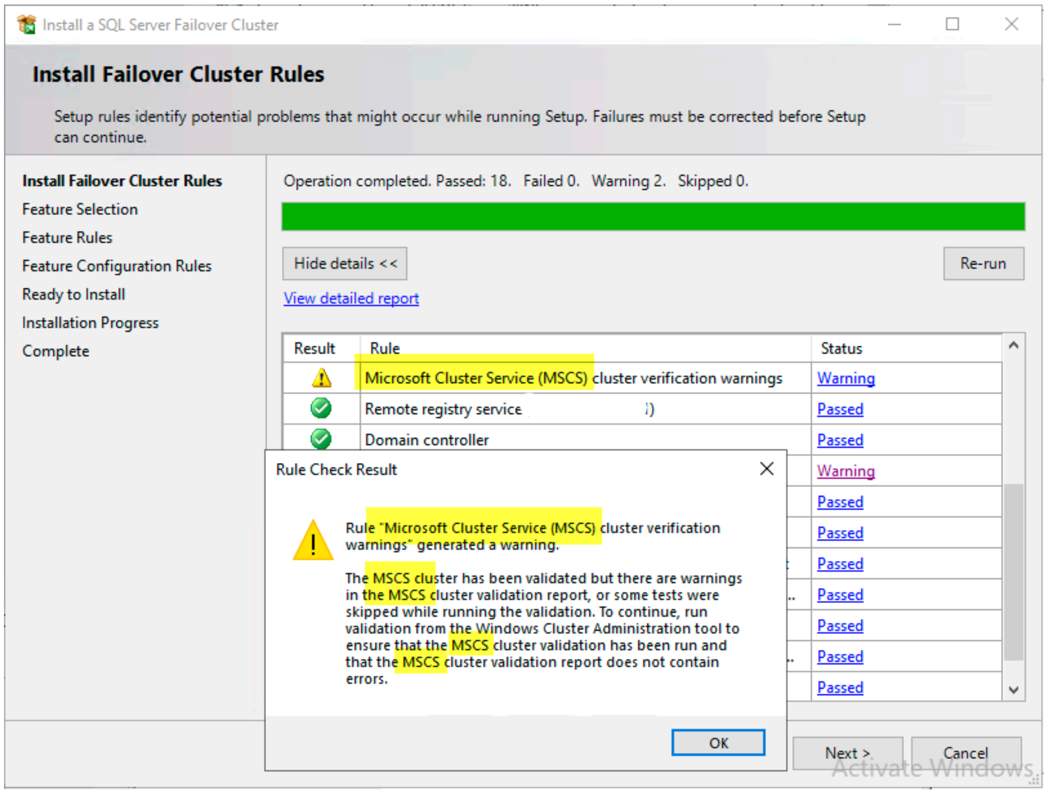
Install SQL Server Management Tools
Launch a download page that provides a link to install SQL Server Management Studio, SQL Server command-line utilities (SQLCMD and BCP), SQL Server PowerShell provider, SQL Server Profiler and Database Tuning Advisor. An internet connection is required to install these tools.

Install SQL Server Data Tools
Launch a download page that provides a link to install SQL Server Data Tools (SSDT). SSDT provides Visual Studio integration including project system support for Microsoft Azure SQL Database, the SQL Server Database Engine, Reporting Services, Analysis Services and Integration Services. An internet connection is required to install SSDT.

New SQL Server failover cluster installation
Launch a wizard to install a single-node SQL Server 2019 failover cluster.

Add node to a SQL Server failover cluster
Launch a wizard to add a node to an existing SQL Server 2019 failover cluster.

The install process wants to remind us that we did ignore some warnings during the WSFC validation test. We know this already, so we will ignore the reminder and proceed with our installation.



Install a SQL Server Failover Cluster

Install Failover Cluster Rules

Setup rules identify potential problems that might occur while running Setup. Failures must be corrected before Setup can continue.

Install Failover Cluster Rules
Feature Selection
Feature Rules
Feature Configuration Rules
Ready to Install
Installation Progress
Complete

Operation completed. Passed: 18. Failed 0. Warning 2. Skipped 0.

Hide details << | Re-run

View detailed report

Result	Rule	Status
⚠	Microsoft Cluster Service (MSCS) cluster verification warnings	Warning
✓	Remote registry service	Passed
✓	Domain controller	Passed
		Warning
		Passed
		Passed
		Passed
		Passed
		Passed
		Passed
		Passed

Rule Check Result

Rule "Microsoft Cluster Service (MSCS) cluster verification warnings" generated a warning.

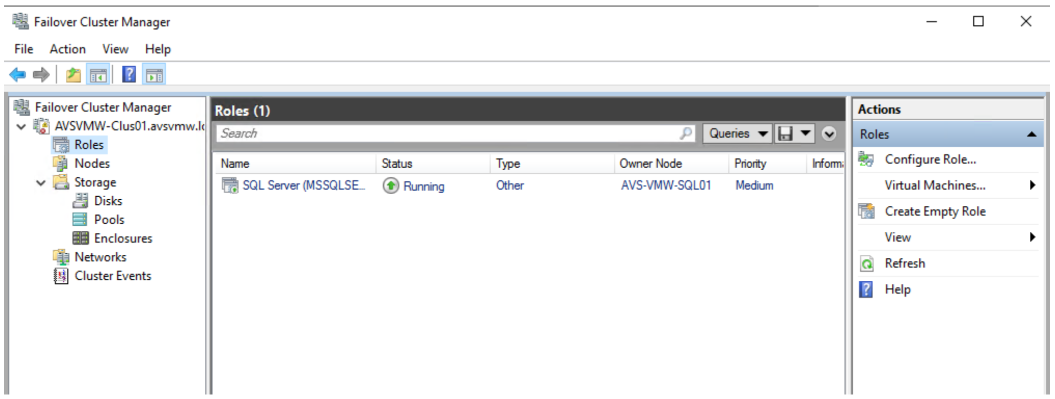
The MSCS cluster has been validated but there are warnings in the MSCS cluster validation report, or some tests were skipped while running the validation. To continue, run validation from the Windows Cluster Administration tool to ensure that the MSCS cluster validation has been run and that the MSCS cluster validation report does not contain errors.

OK

Next > | Cancel

Activate Windows

- After successfully installing SQL Server on SQL01, we can see in Cluster Manager that **SQL Server (MSSQLServer)** has become a clustered role in WSFC.



Name	Status	Type	Owner Node	Priority	Informa
SQL Server (MSSQLSE...	Running	Other	AVS-VMW-SQL01	Medium	

We can now install SQL Server on the second Node (SQL02), joining it in as a part of the SQL Server cluster we just installed on node 1 (SQL01). But, before we do so, we have some housekeeping tasks to take care of. One is related to improving disk IO for Microsoft SQL Server workloads in vSphere-based hybrid cloud environments, and the other is maintaining availability and resilience for WSFC-based clustered resources generally.

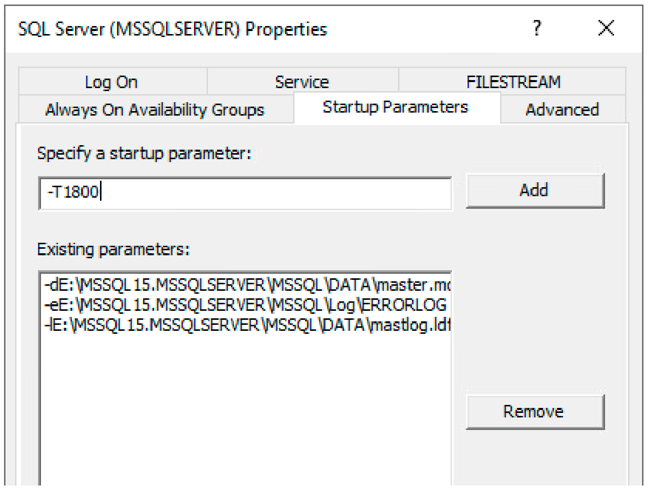
Enable (Persistent) Trace Flag 1800

Disk IO misalignment is a very common cause of severe suboptimal performance of Microsoft SQL Server queries, synchronization and transactions. Although Microsoft has long released patches to address the specific Sector misalignment issues, an administrative action is still required in order for the fixes to apply to a given SQL Server instance. Please see *KB3009974 - FIX: Slow synchronization when disks have different sector sizes for primary and secondary replica log files in SQL Server AG and Logshipping environments (microsoft.com)* for more detailed information.

So, let's go ahead and do this for our environment.

The TraceFlag will be global and persistent (it needs to remain in place, even after a service restart or server reboot), so we will do this inside the SQL Server Configuration Manager (SSCM).

- Open SSCM and navigate to the **Startup Parameters** tab.
- Type in **-T1800** (TraceFlags trace flags are preceded by "-T"), then click **Add**.


SQL Server (MSSQLSERVER) Properties
?
Log On
Service
FILESTREAM
Always On Availability Groups
Startup Parameters
Advanced
Add
Remove

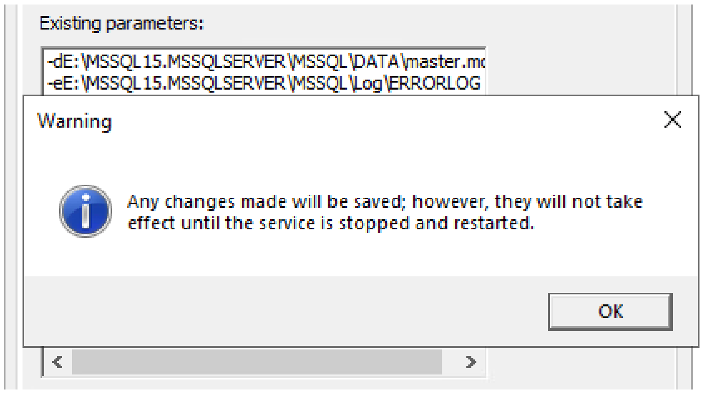
Specify a startup parameter:

-T1800

Existing parameters:

```
-dE:\MSSQL15.MSSQLSERVER\MSSQL\DATA\master.md
-eE:\MSSQL15.MSSQLSERVER\MSSQL\Log\ERRORLOG
-lE:\MSSQL15.MSSQLSERVER\MSSQL\DATA\mastlog.ldf
```

- Click **OK**.



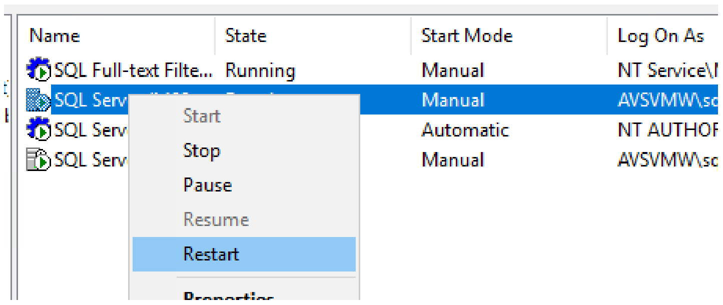
Existing parameters:

```
-dE:\MSSQL15.MSSQLSERVER\MSSQL\DATA\master.md
-eE:\MSSQL15.MSSQLSERVER\MSSQL\Log\ERRORLOG
```

Warning

Any changes made will be saved; however, they will not take effect until the service is stopped and restarted.

Yes, yes, we have to restart the service for this to take effect, so let's do so now.

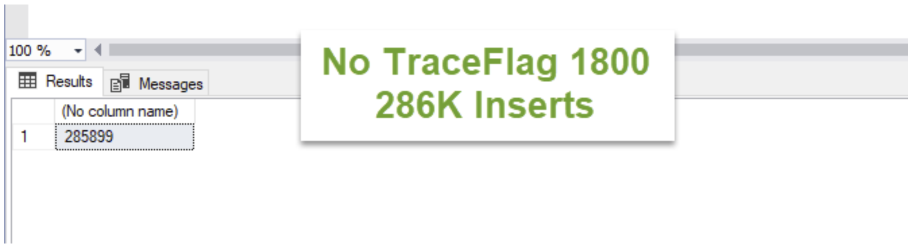


Name	State	Start Mode	Log On As
SQL Full-text Filte...	Running	Manual	NT Service\I
SQL Serv		Manual	AVSVMW\sc
SQL Serv		Automatic	NT AUTHOF
SQL Serv		Manual	AVSVMW\sc

That's it.

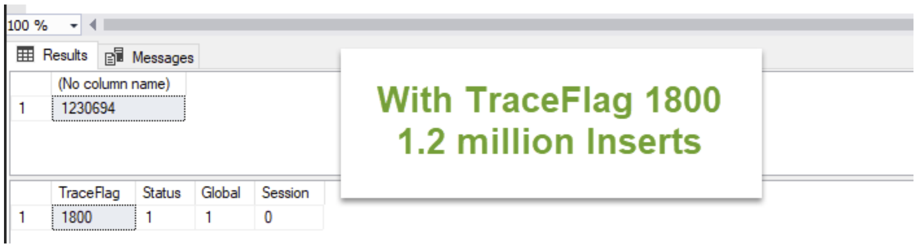
But, was all that worth the effort? Let's take a look at *before* and *after*.

Here is a sample insert job we ran on a server without the trace flag:



	(No column name)
1	285899

Here is the same job, with trace flag 1800 enabled on the same server:

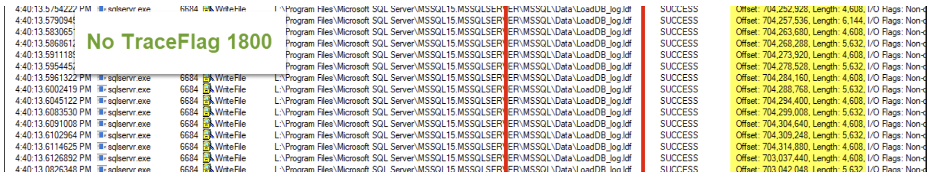


	(No column name)
1	1230694

	TraceFlag	Status	Global	Session
1	1800	1	1	0

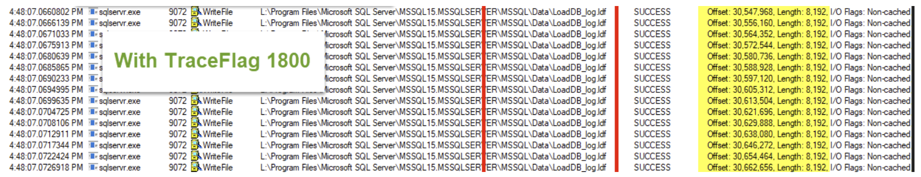
Using a tool like Procmon, you can get a peek into the root cause of the performance bottleneck and see how it resolves after the TraceFlag trace flag is added.

Here is our IO and block size pattern prior to enabling –T1800 (the yellow highlight):



```
4:40:13.5754222 PM  sqlservr.exe  6684  WriteFile  L:\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\Data\LoadDB_log.ldf  SUCCESS  Offset: 704,252,928, Length: 4,608, I/O Flags: Non-c
4:40:13.5790945                                    Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\Data\LoadDB_log.ldf  SUCCESS  Offset: 704,257,536, Length: 6,144, I/O Flags: Non-c
4:40:13.5830651                                    Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\Data\LoadDB_log.ldf  SUCCESS  Offset: 704,263,680, Length: 4,608, I/O Flags: Non-c
4:40:13.5868612                                    Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\Data\LoadDB_log.ldf  SUCCESS  Offset: 704,268,288, Length: 5,632, I/O Flags: Non-c
4:40:13.5911189                                    Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\Data\LoadDB_log.ldf  SUCCESS  Offset: 704,273,920, Length: 4,608, I/O Flags: Non-c
4:40:13.5954452                                    Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\Data\LoadDB_log.ldf  SUCCESS  Offset: 704,278,528, Length: 5,632, I/O Flags: Non-c
4:40:13.5961322 PM  sqlservr.exe  6684  WriteFile  L:\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\Data\LoadDB_log.ldf  SUCCESS  Offset: 704,284,160, Length: 4,608, I/O Flags: Non-c
4:40:13.6002419 PM  sqlservr.exe  6684  WriteFile  L:\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\Data\LoadDB_log.ldf  SUCCESS  Offset: 704,288,768, Length: 5,632, I/O Flags: Non-c
4:40:13.6045122 PM  sqlservr.exe  6684  WriteFile  L:\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\Data\LoadDB_log.ldf  SUCCESS  Offset: 704,294,400, Length: 4,608, I/O Flags: Non-c
4:40:13.6083530 PM  sqlservr.exe  6684  WriteFile  L:\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\Data\LoadDB_log.ldf  SUCCESS  Offset: 704,299,008, Length: 5,632, I/O Flags: Non-c
4:40:13.6091008 PM  sqlservr.exe  6684  WriteFile  L:\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\Data\LoadDB_log.ldf  SUCCESS  Offset: 704,304,640, Length: 4,608, I/O Flags: Non-c
4:40:13.6102964 PM  sqlservr.exe  6684  WriteFile  L:\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\Data\LoadDB_log.ldf  SUCCESS  Offset: 704,309,248, Length: 5,632, I/O Flags: Non-c
4:40:13.6114625 PM  sqlservr.exe  6684  WriteFile  L:\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\Data\LoadDB_log.ldf  SUCCESS  Offset: 704,314,880, Length: 4,608, I/O Flags: Non-c
4:40:13.6126892 PM  sqlservr.exe  6684  WriteFile  L:\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\Data\LoadDB_log.ldf  SUCCESS  Offset: 703,037,440, Length: 4,608, I/O Flags: Non-c
4:40:13.0826348 PM  sqlservr.exe  6684  WriteFile  L:\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\Data\LoadDB_log.ldf  SUCCESS  Offset: 703,042,048, Length: 5,632, I/O Flags: Non-c
```

Here is how it improved and stabilized with -T1800:



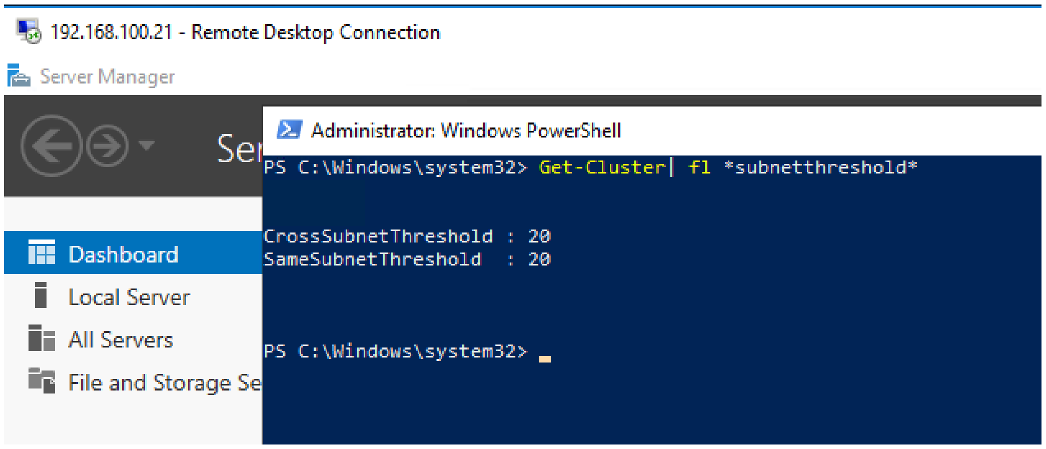
```
4:48:07.0660802 PM  sqlservr.exe  9072  WriteFile  L:\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\Data\LoadDB_log.ldf  SUCCESS  Offset: 30,547,968, Length: 8,192, I/O Flags: Non-cached
4:48:07.0666139 PM  sqlservr.exe  9072  WriteFile  L:\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\Data\LoadDB_log.ldf  SUCCESS  Offset: 30,556,160, Length: 8,192, I/O Flags: Non-cached
4:48:07.0671033 PM  s                                            icrosoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\Data\LoadDB_log.ldf  SUCCESS  Offset: 30,564,352, Length: 8,192, I/O Flags: Non-cached
4:48:07.0675913 PM  s                                            icrosoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\Data\LoadDB_log.ldf  SUCCESS  Offset: 30,572,544, Length: 8,192, I/O Flags: Non-cached
4:48:07.0680639 PM  s                                            icrosoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\Data\LoadDB_log.ldf  SUCCESS  Offset: 30,580,736, Length: 8,192, I/O Flags: Non-cached
4:48:07.0685865 PM  s                                            icrosoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\Data\LoadDB_log.ldf  SUCCESS  Offset: 30,588,928, Length: 8,192, I/O Flags: Non-cached
4:48:07.0690233 PM  s                                            icrosoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\Data\LoadDB_log.ldf  SUCCESS  Offset: 30,597,120, Length: 8,192, I/O Flags: Non-cached
4:48:07.0694995 PM  s                                            icrosoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\Data\LoadDB_log.ldf  SUCCESS  Offset: 30,605,312, Length: 8,192, I/O Flags: Non-cached
4:48:07.0699635 PM  sqlservr.exe  9072  WriteFile  L:\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\Data\LoadDB_log.ldf  SUCCESS  Offset: 30,613,504, Length: 8,192, I/O Flags: Non-cached
4:48:07.0704725 PM  sqlservr.exe  9072  WriteFile  L:\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\Data\LoadDB_log.ldf  SUCCESS  Offset: 30,621,696, Length: 8,192, I/O Flags: Non-cached
4:48:07.0708106 PM  sqlservr.exe  9072  WriteFile  L:\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\Data\LoadDB_log.ldf  SUCCESS  Offset: 30,629,888, Length: 8,192, I/O Flags: Non-cached
4:48:07.0712911 PM  sqlservr.exe  9072  WriteFile  L:\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\Data\LoadDB_log.ldf  SUCCESS  Offset: 30,638,080, Length: 8,192, I/O Flags: Non-cached
4:48:07.0717344 PM  sqlservr.exe  9072  WriteFile  L:\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\Data\LoadDB_log.ldf  SUCCESS  Offset: 30,646,272, Length: 8,192, I/O Flags: Non-cached
4:48:07.0722424 PM  sqlservr.exe  9072  WriteFile  L:\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\Data\LoadDB_log.ldf  SUCCESS  Offset: 30,654,464, Length: 8,192, I/O Flags: Non-cached
4:48:07.0726918 PM  sqlservr.exe  9072  WriteFile  L:\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\Data\LoadDB_log.ldf  SUCCESS  Offset: 30,662,656, Length: 8,192, I/O Flags: Non-cached
```

Adjust WSFC Cluster Heartbeat Thresholds:

One of the standard configuration checks you'd want to perform after completing your clustered SQL Server installation on AVS is the **Cluster Heartbeat Threshold** setting. You can do this by issuing the following PowerShell command:

*Get-Cluster | fl *subnetthreshold**

You should get an output similar to that shown below:


192.168.100.21 - Remote Desktop Connection
Server Manager
Ser
Administrator: Windows PowerShell
Dashboard
Local Server
All Servers
File and Storage Se

```
PS C:\Windows\system32> Get-Cluster| fl *subnetthreshold*


CrossSubnetThreshold : 20
SameSubnetThreshold  : 20



PS C:\Windows\system32> _
```

Why is this necessary? It's a long story, but long documented in many places, including in *VMware's Best Practices for Virtualizing Microsoft SQL Server* guides. Here's a brief overview:

Clustered Windows nodes exchange periodic heartbeats with each other. This is how they verify that their peers are available, responsive and still participating in the cluster. They send these heartbeats every second. Historically, if one node fails to respond to five (for nodes in the same network subnet) or ten (for nodes in different subnets) consecutive heartbeat probes, its peers will consider it unresponsive and will be compelled to try and take over whatever clustered resource is hosted on the non-responsive node. This means that a node has a grace period of 5 or 10 seconds to respond to heartbeat probes before it's deemed to no longer be fit to own a clustered resource.

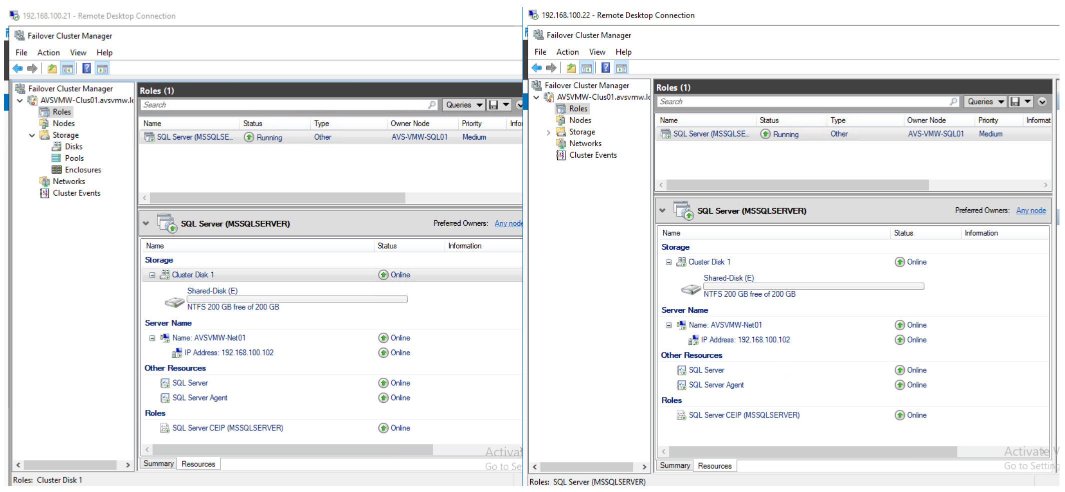
These numbers were considered good enough until freezing (or quiescing) a Windows OS instance became a feature of Windows and a standard practice for many operations (e.g., in-Guest backup). When a clustered Windows OS instance is quiesced for longer than the tolerable threshold, unexpected clustered resource failover events begin to occur.

To help mitigate this potential for unintended failovers, Microsoft recommended that customers adjust the threshold. Microsoft has now since increased the default to 20, for both **SameSubnetThreshold** and **CrossSubnetThreshold** for more modern Windows versions.

In AVS (indeed, in all VMware vSphere platforms), when you perform a vMotion operation on a Windows VM, the VM will be quiesced at a point during the process. Although vMotion tries to exist the quiesced state as quickly as possible, some external factors could cause the process to take longer than 5 or 10 seconds in some versions of Windows. For clustered workloads, this is not desirable. This is why you must check and adjust the thresholds. If you need to adjust the threshold, use the following PowerShell commands:

- **(get-cluster).SameSubnetThreshold = 20**
- **(get-cluster).CrossSubnetThreshold = 20**
- **(get-cluster).RouteHistoryLength = 40**

We now have a 2-node SQL Server instance configured in FCI clustering mode in our AVS. Let see what it looks like.



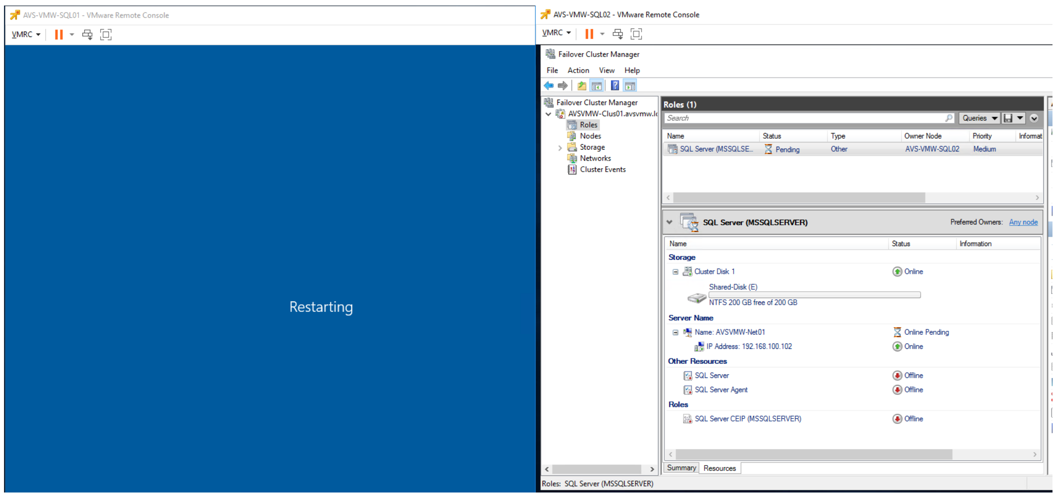
Roles (1)

Name	Status	Type	Owner Node	Priority
SQL Server (MSSQLSE...	Running	Other	AVS-VMW-SQL01	Medium

SQL Server (MSSQLSERVER) Preferred Owners: Any node

Name	Status	Information
Storage		
Cluster Disk 1	Online	
Shared-Disk (E) NTFS 200 GB free of 200 GB		
Server Name		
Name: AVSVMW-Net01	Online	
IP Address: 192.168.100.102	Online	
Other Resources		
SQL Server	Online	
SQL Server Agent	Online	
Roles		
SQL Server CEIP (MSSQLSERVER)	Online	

Node 1 (SQL01) currently owns the resource. Let's see what happens if SQL01 were to suddenly become unavailable.



Roles (1)

Name	Status	Type	Owner Node	Priority
SQL Server (MSSQLSE...	Pending	Other	AVS-VMW-SQL02	Medium

SQL Server (MSSQLSERVER) Preferred Owners: Any node

Name	Status	Information
Storage		
Cluster Disk 1	Online	
Shared-Disk (E) NTFS 200 GB free of 200 GB		
Server Name		
Name: AVSVMW-Net01	Online Pending	
IP Address: 192.168.100.102	Online	
Other Resources		
SQL Server	Offline	
SQL Server Agent	Offline	
Roles		
SQL Server CEIP (MSSQLSERVER)	Offline	

As expected, the resource automatically fails over to node 2 (SQL02). Nothing to see here.

Installing and Configuring Microsoft SQL Server Always On Availability Group on AVS

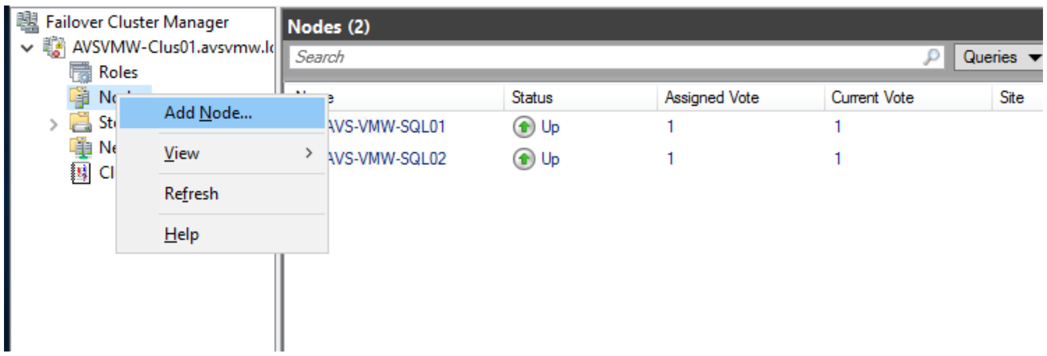
Unlike FCI, installing and configuring Microsoft SQL Server Always On availability group (AG) on Azure VMware Solution is a much more simplified process. This is chiefly because shared-disk SCSI3 PR command and disk ownership arbitration among two or more clustered VMs is the major challenge with successfully configuring Windows Server Failover Clustering in a VMware vSphere-based platform such as AVS. In AG, the nodes do not share disks or resources. Each node has ownership of its own set of resources.

We will now describe the tasks and activities involved in successfully creating and configuring a sample Availability Group cluster in our AVS environment. We will attempt to make this section more interesting by adding the AG configuration on top of our existing FCI configuration. This is a fairly common configuration in production environment and it's our hope that, in using this example, administrators will get a more detailed picture of the various configuration options available in clustering Microsoft SQL Server on AVS.

– **NOTE:** As usual, we expect you are familiar with the administrative tasks and preparations required for some of the tasks we show in this example. We will not be describing this in detail, except to the extent that the tasks are either unique to AVS or help paint a clearer picture.

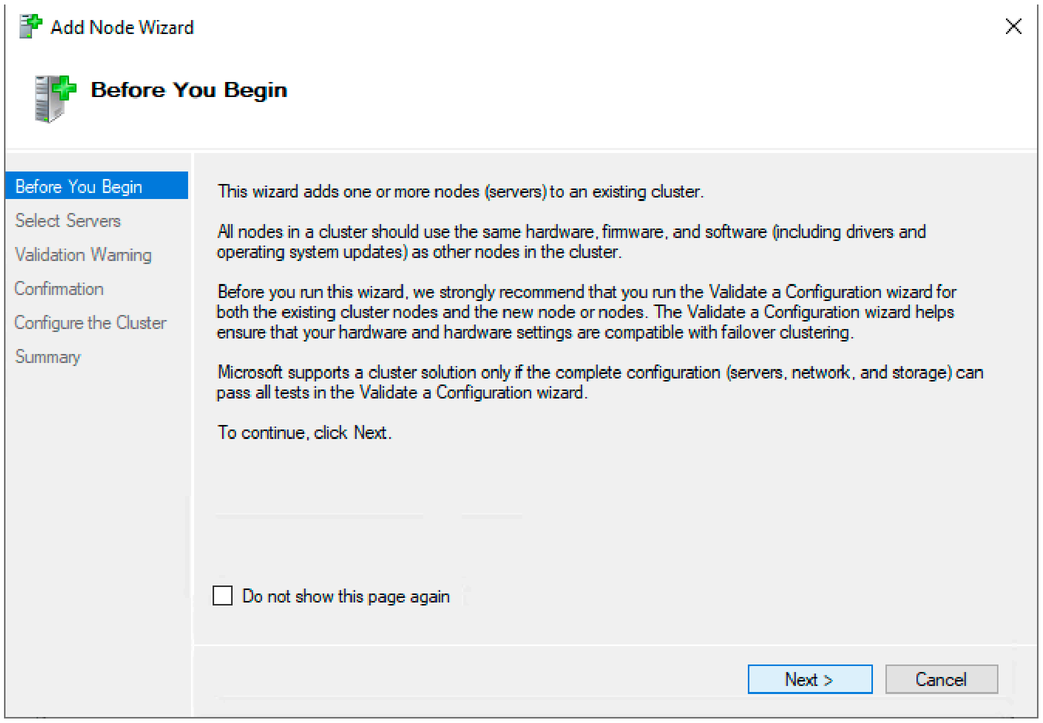
  For example, we have added a separate, non-shared VMDK to SQL01 which will contain the data and files for the Database that we're going to configure in AG. We have also added a similar (but separate and unshared) VMDK to SQL03 for the same purpose.

We will now proceed by adding a third MS SQL Server node (i.e., AVS-VMW-SQL03, hereafter simply referred to as SQL03) to the existing WSFC cluster we created in the previous exercise.



Name	Status	Assigned Vote	Current Vote	Site
AVS-VMW-SQL01	Up	1	1	
AVS-VMW-SQL02	Up	1	1	

- Using Cluster Manager, expand the cluster, right click on **Nodes**, and choose **Add Node**.



This wizard adds one or more nodes (servers) to an existing cluster.

All nodes in a cluster should use the same hardware, firmware, and software (including drivers and operating system updates) as other nodes in the cluster.

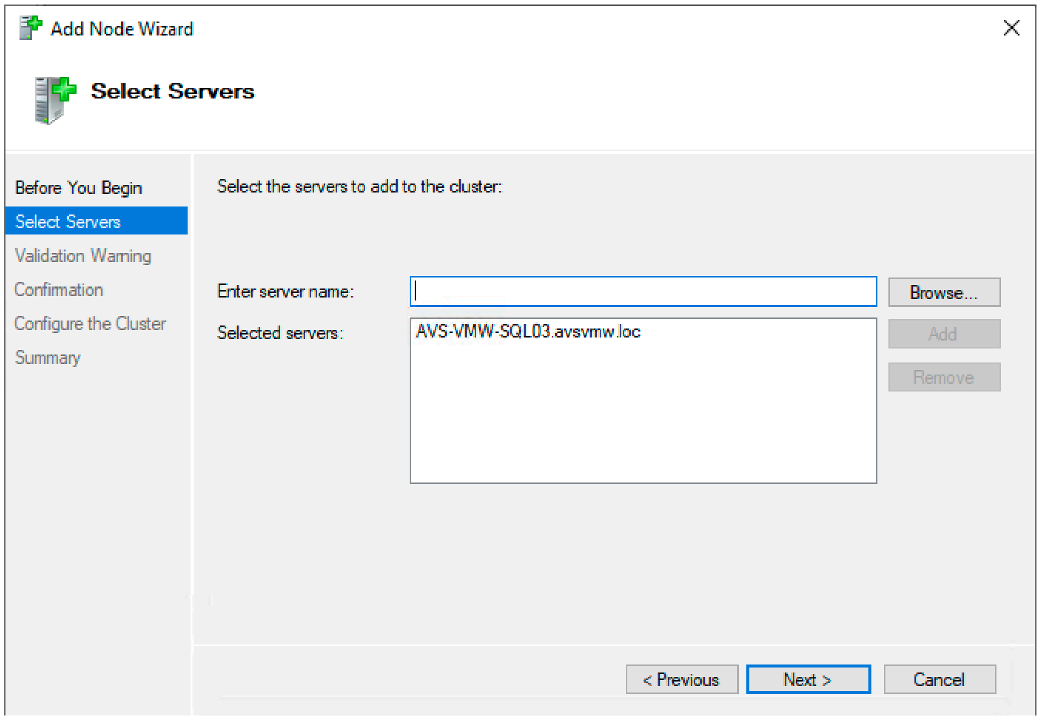
Before you run this wizard, we strongly recommend that you run the Validate a Configuration wizard for both the existing cluster nodes and the new node or nodes. The Validate a Configuration wizard helps ensure that your hardware and hardware settings are compatible with failover clustering.

Microsoft supports a cluster solution only if the complete configuration (servers, network, and storage) can pass all tests in the Validate a Configuration wizard.

To continue, click Next.

☐ Do not show this page again

- Click **Next**.

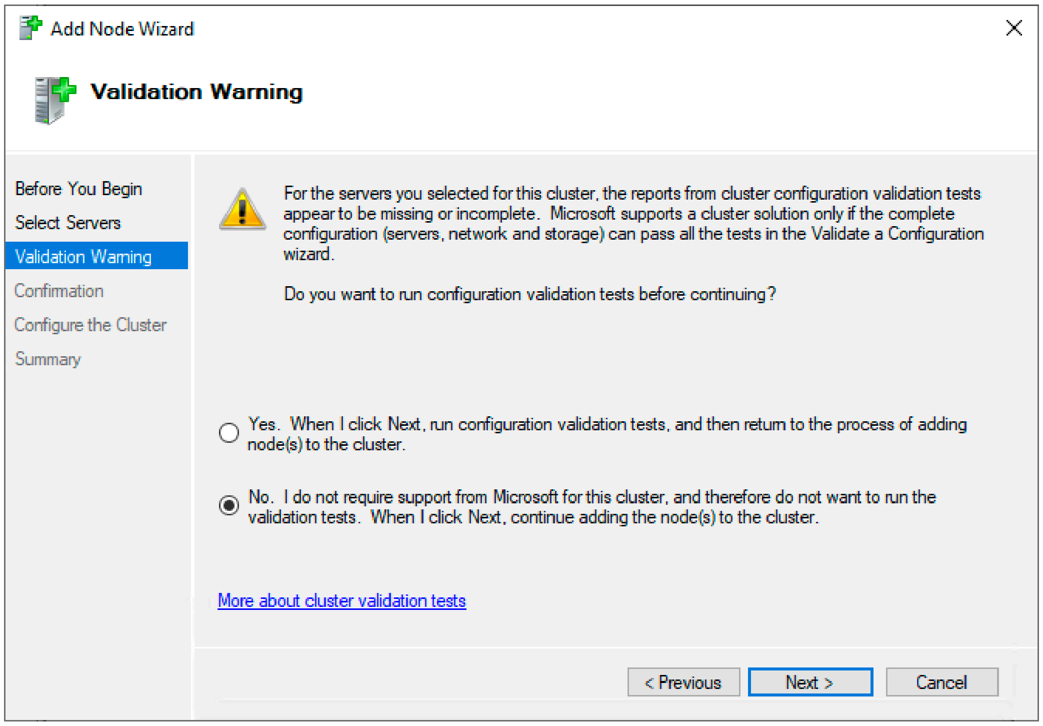


Select the servers to add to the cluster:

Enter server name:

Selected servers: AVS-VMW-SQL03.avsvmw.loc

- We'll add SQL03 and click **Next**.



For the servers you selected for this cluster, the reports from cluster configuration validation tests appear to be missing or incomplete. Microsoft supports a cluster solution only if the complete configuration (servers, network and storage) can pass all the tests in the Validate a Configuration wizard.

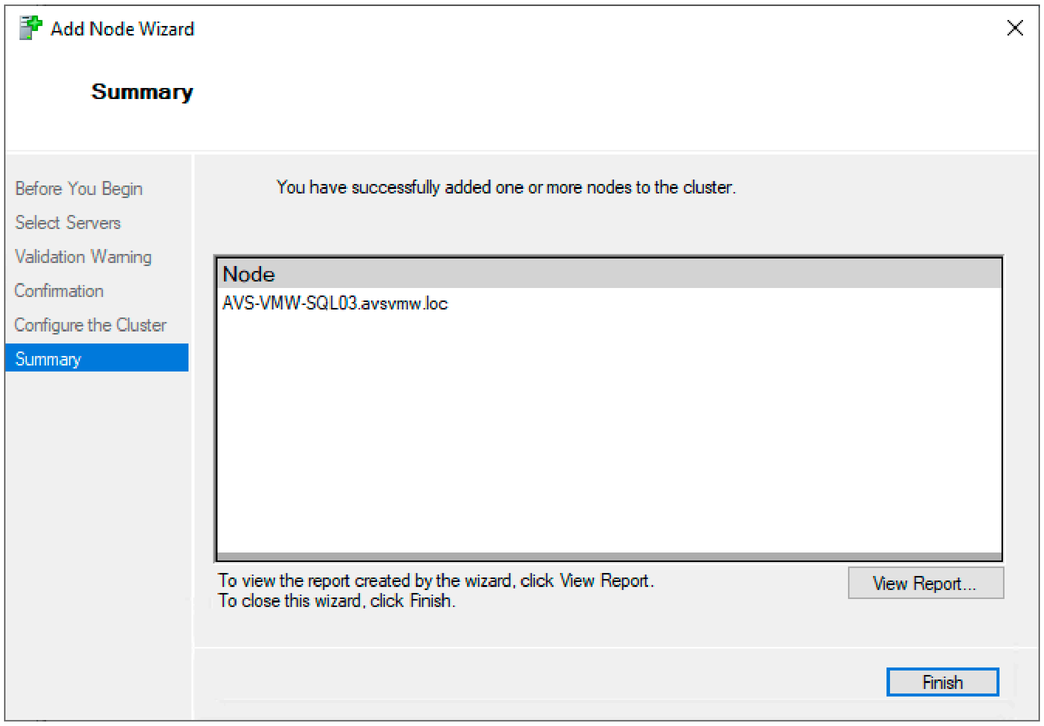
Do you want to run configuration validation tests before continuing?

Yes. When I click Next, run configuration validation tests, and then return to the process of adding node(s) to the cluster.

[✓ No. I do not require support from Microsoft for this cluster, and therefore do not want to run the validation tests. When I click Next, continue adding the node(s) to the cluster.]

More about cluster validation tests

• We've already run the validation test and passed, so, let's skip it this time.

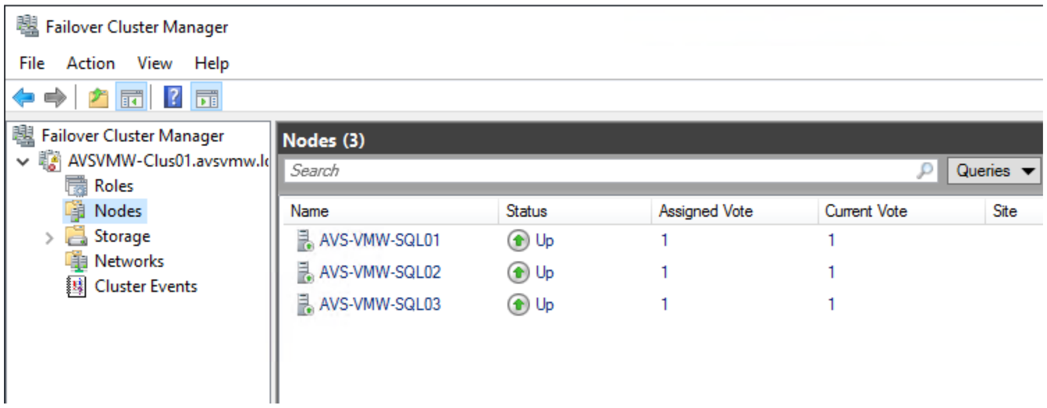


You have successfully added one or more nodes to the cluster.

Node
AVS-VMW-SQL03.avsvmw.loc

To view the report created by the wizard, click View Report.
To close this wizard, click Finish.

• One more click and we are done with adding SQL03 to our WSFC cluster. Here is what we have now:

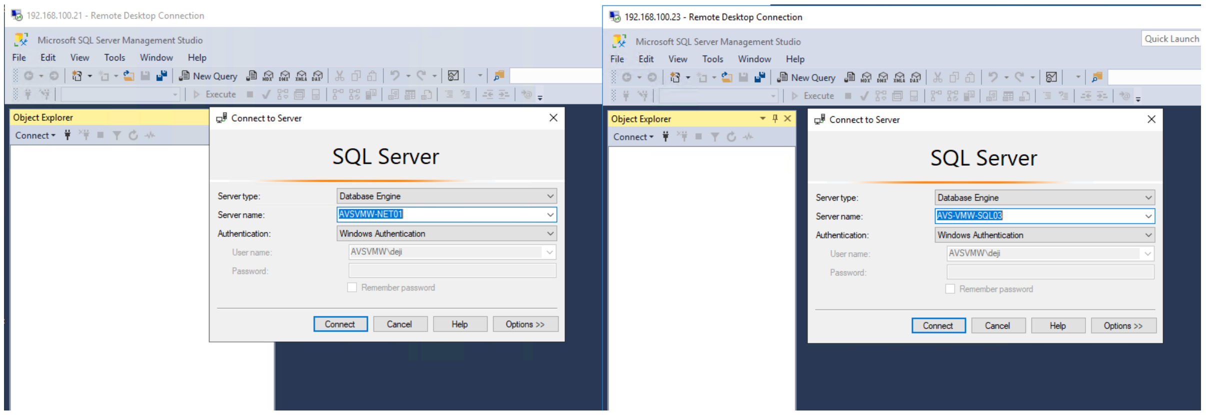


Nodes (3)

Name	Status	Assigned Vote	Current Vote	Site
AVS-VMW-SQL01	Up	1	1	
AVS-VMW-SQL02	Up	1	1	
AVS-VMW-SQL03	Up	1	1	

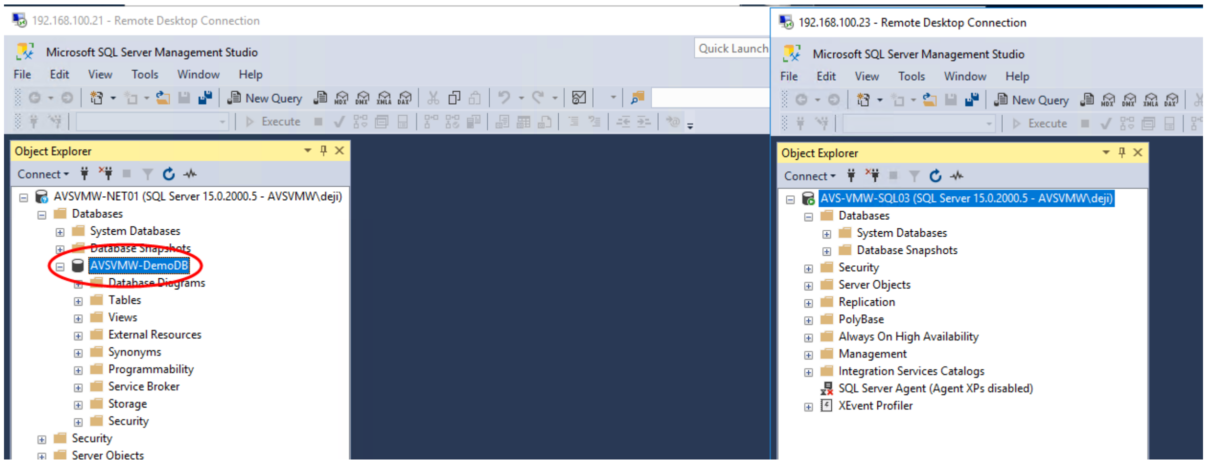
Now, we move onto configuring a Database for Always On AG.

• Let's connect to the SQL instances on SQL01 and SQL03.


192.168.100.21 - Remote Desktop Connection
Microsoft SQL Server Management Studio
File Edit View Tools Window Help
New Query
Execute
Object Explorer
Connect
Connect to Server
SQL Server
Connect
Cancel
Help
Options >>
192.168.100.23 - Remote Desktop Connection
Quick Launch

Server type: Database Engine
Server name: AVSVMW-NET01
Authentication: Windows Authentication
User name: AVSVMW\deji
Password:
Remember password

Server type: Database Engine
Server name: AVS-VMW-SQL03
Authentication: Windows Authentication
User name: AVSVMW\deji
Password:
Remember password


192.168.100.21 - Remote Desktop Connection
Quick Launch
Microsoft SQL Server Management Studio
File Edit View Tools Window Help
New Query
Execute
Object Explorer
Connect
192.168.100.23 - Remote Desktop Connection

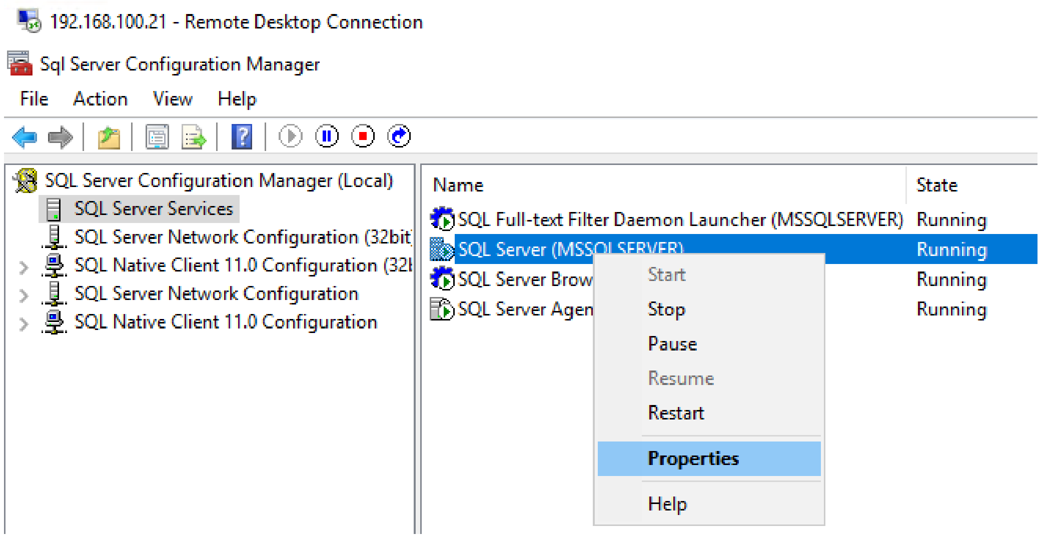
AVSVMW-NET01 (SQL Server 15.0.2000.5 - AVSVMW\deji)
- Databases
  - System Databases
  - Database Snapshots
  - AVSVMW-DemoDB
    - Database Diagrams
    - Tables
    - Views
    - External Resources
    - Synonyms
    - Programmability
    - Service Broker
    - Storage
    - Security
- Security
- Server Objects

AVS-VMW-SQL03 (SQL Server 15.0.2000.5 - AVSVMW\deji)
- Databases
  - System Databases
  - Database Snapshots
- Security
- Server Objects
- Replication
- PolyBase
- Always On High Availability
- Management
- Integration Services Catalogs
- SQL Server Agent (Agent XPs disabled)
- XEvent Profiler

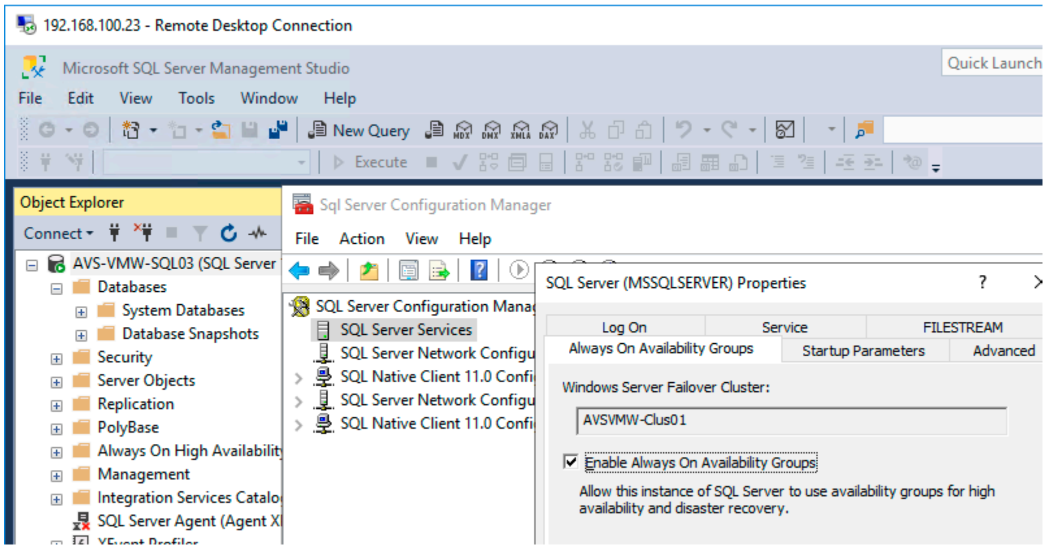
• There is currently one Database on SQL01. This is the Database in our default instance which is configured in FCI and paired with SQL02. There is currently no Database on SQL03 because this node is not a part of the FCI configuration.

We will now configure our target nodes to participate in Always On Availability Group configuration.

- On each node, click the **SQL Server Services** properties in SQL Server Configuration Manager.


192.168.100.21 - Remote Desktop Connection
Sql Server Configuration Manager
File Action View Help
SQL Server Configuration Manager (Local)
SQL Server Services
SQL Server Network Configuration (32bit
SQL Native Client 11.0 Configuration (32
SQL Server Network Configuration
SQL Native Client 11.0 Configuration
Start
Stop
Pause
Resume
Restart
Properties
Help

Name	State
SQL Full-text Filter Daemon Launcher (MSSQLSERVER)	Running
SQL Server (MSSQLSERVER)	Running
SQL Server Brow	Running
SQL Server Agen	Running


192.168.100.23 - Remote Desktop Connection
Microsoft SQL Server Management Studio
Quick Launch
File Edit View Tools Window Help
New Query
Execute
Object Explorer
Connect
AVS-VMW-SQL03 (SQL Server
Databases
System Databases
Database Snapshots
Security
Server Objects
Replication
PolyBase
Always On High Availabilit
Management
Integration Services Catalo
SQL Server Agent (Agent X
Sql Server Configuration Manager
File Action View Help
SQL Server Configuration Mana
SQL Server Services
SQL Server Network Configu
SQL Native Client 11.0 Confi
SQL Server Network Configu
SQL Native Client 11.0 Confi
SQL Server (MSSQLSERVER) Properties
Log On
Service
FILESTREAM
Always On Availability Groups
Startup Parameters
Advanced

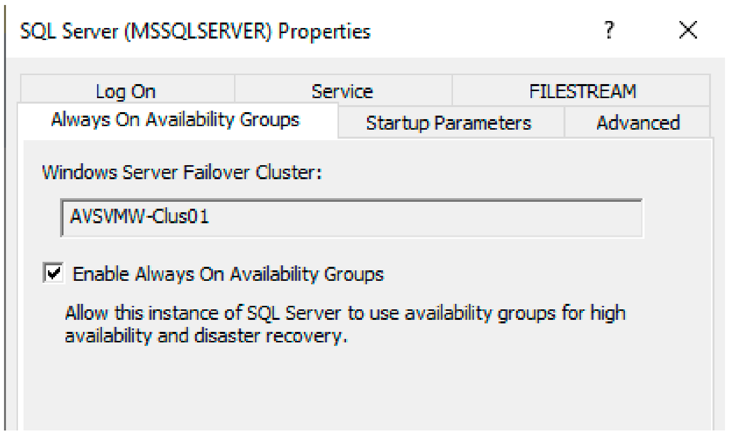
Windows Server Failover Cluster:

AVSVMW-Clus01

☑ Enable Always On Availability Groups

Allow this instance of SQL Server to use availability groups for high availability and disaster recovery.

• Check the **Enable Always On Availability Groups** box on the **Always On Availability Groups** tab.


SQL Server (MSSQLSERVER) Properties

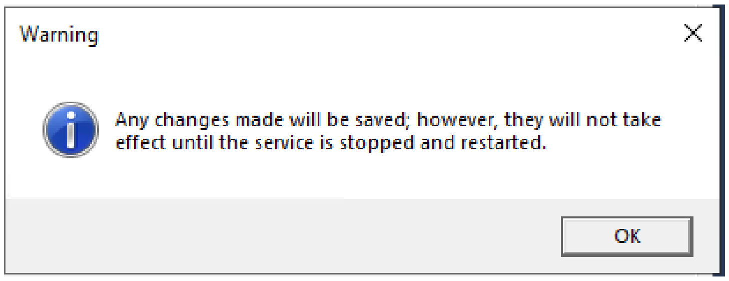
Windows Server Failover Cluster:

AVSVMW-Clus01

☑ Enable Always On Availability Groups

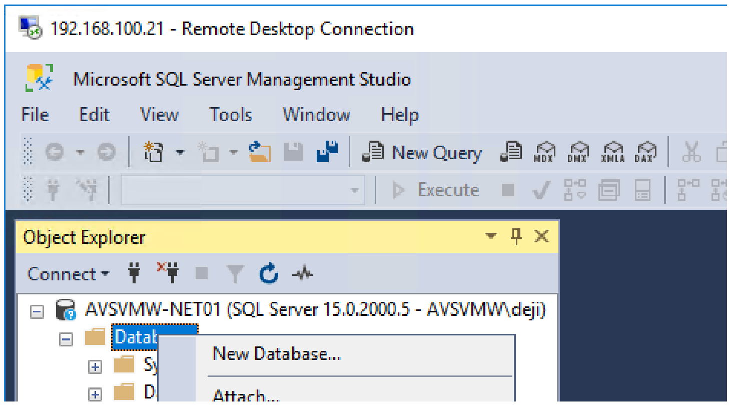
Allow this instance of SQL Server to use availability groups for high availability and disaster recovery.

• Click **OK** to confirm that we know what we are doing.


Warning

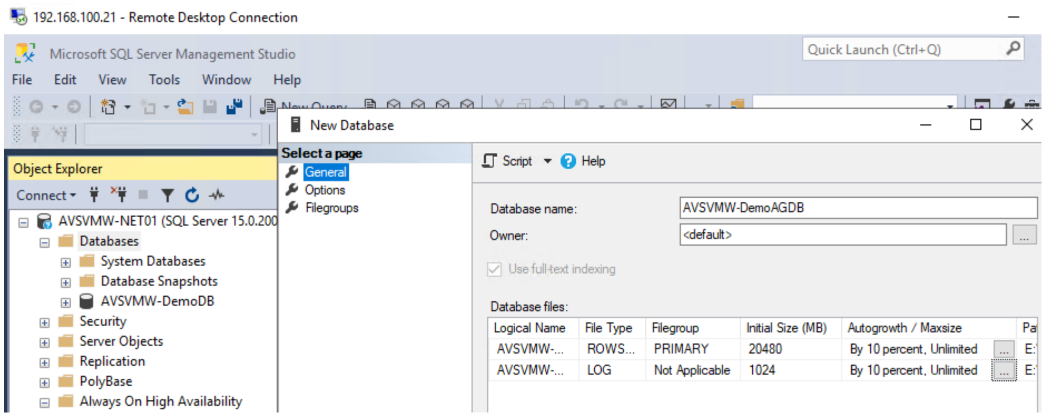
Any changes made will be saved; however, they will not take effect until the service is stopped and restarted.

• Now, we create the Database we're going to use for AG on SQL01.


192.168.100.21 - Remote Desktop Connection

Microsoft SQL Server Management Studio

AVSVMW-NET01 (SQL Server 15.0.2000.5 - AVSVMW\deji)

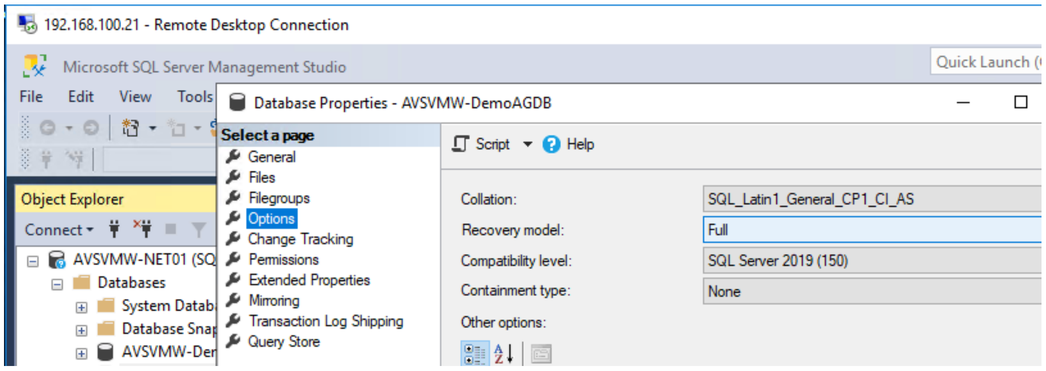


Database name: AVSVMW-DemoAGDB
Owner: <default>
Use full-text indexing

Database files:

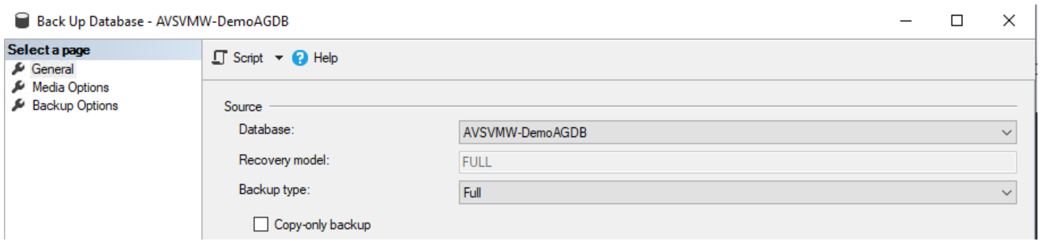
Logical Name	File Type	Filegroup	Initial Size (MB)	Autogrowth / Maxsize	Pa
AVSVMW-...	ROWS...	PRIMARY	20480	By 10 percent, Unlimited	E:
AVSVMW-...	LOG	Not Applicable	1024	By 10 percent, Unlimited	E:

- For our purposes, we will set it to **"Full"** Recovery model.

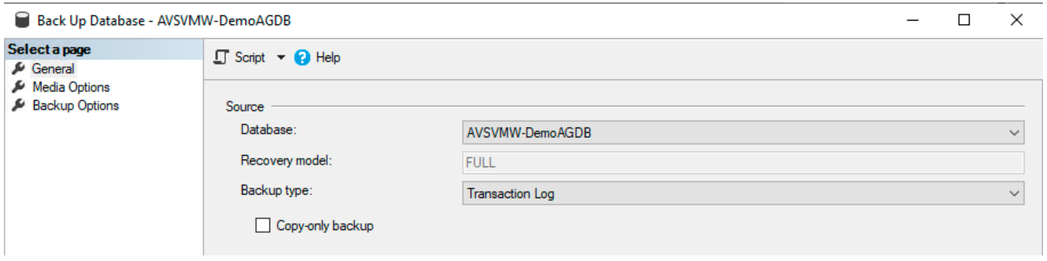


Collation: SQL_Latin1_General_CP1_CI_AS
Recovery model: Full
Compatibility level: SQL Server 2019 (150)
Containment type: None
Other options:

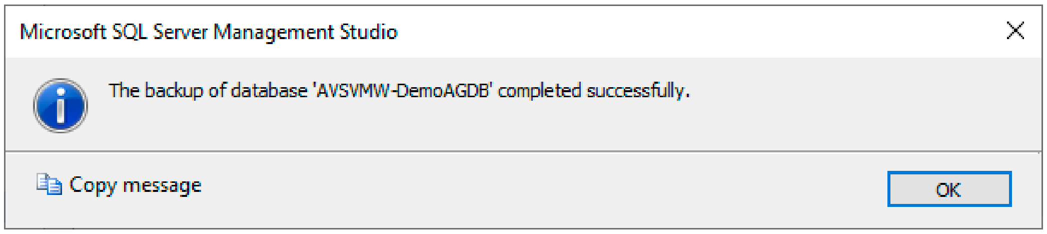
- To help speed up the process, we will take a backup of both the Database and Transaction Logs files.



Source
Database: AVSVMW-DemoAGDB
Recovery model: FULL
Backup type: Full
Copy-only backup

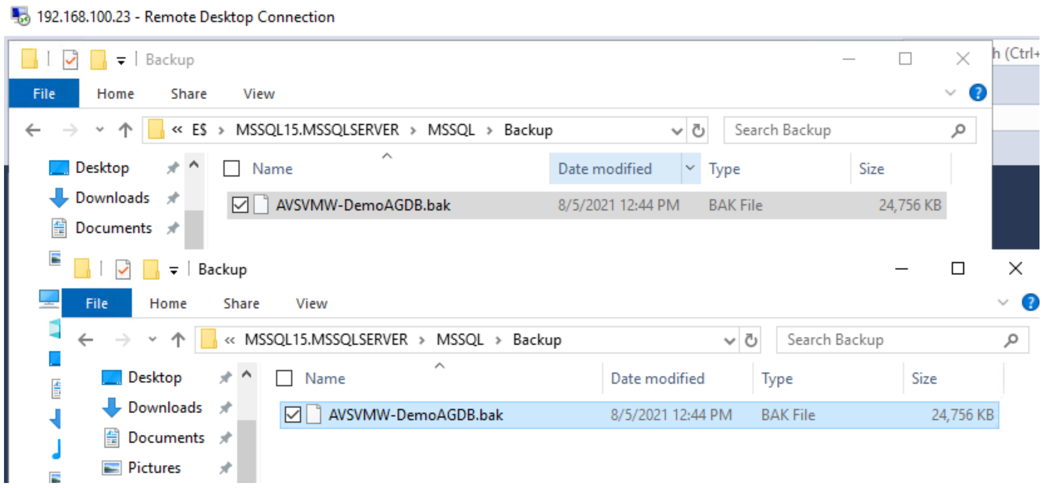


Source
Database: AVSVMW-DemoAGDB
Recovery model: FULL
Backup type: Transaction Log
Copy-only backup


Microsoft SQL Server Management Studio

The backup of database 'AVSVMW-DemoAGDB' completed successfully.

• We'll copy the backup file manually from SQL01 to SQL03.


192.168.100.23 - Remote Desktop Connection

Backup

« E$ › MSSQL15.MSSQLSERVER › MSSQL › Backup

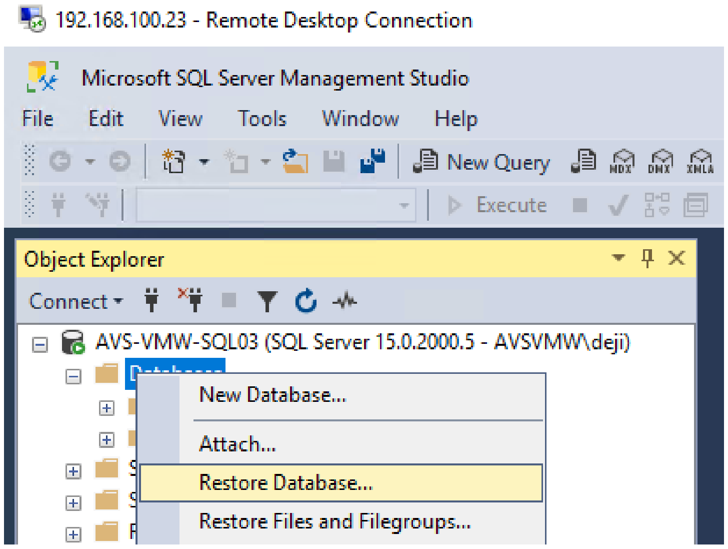
Name	Date modified	Type	Size
AVSVMW-DemoAGDB.bak	8/5/2021 12:44 PM	BAK File	24,756 KB

Backup

« MSSQL15.MSSQLSERVER › MSSQL › Backup

Name	Date modified	Type	Size
AVSVMW-DemoAGDB.bak	8/5/2021 12:44 PM	BAK File	24,756 KB

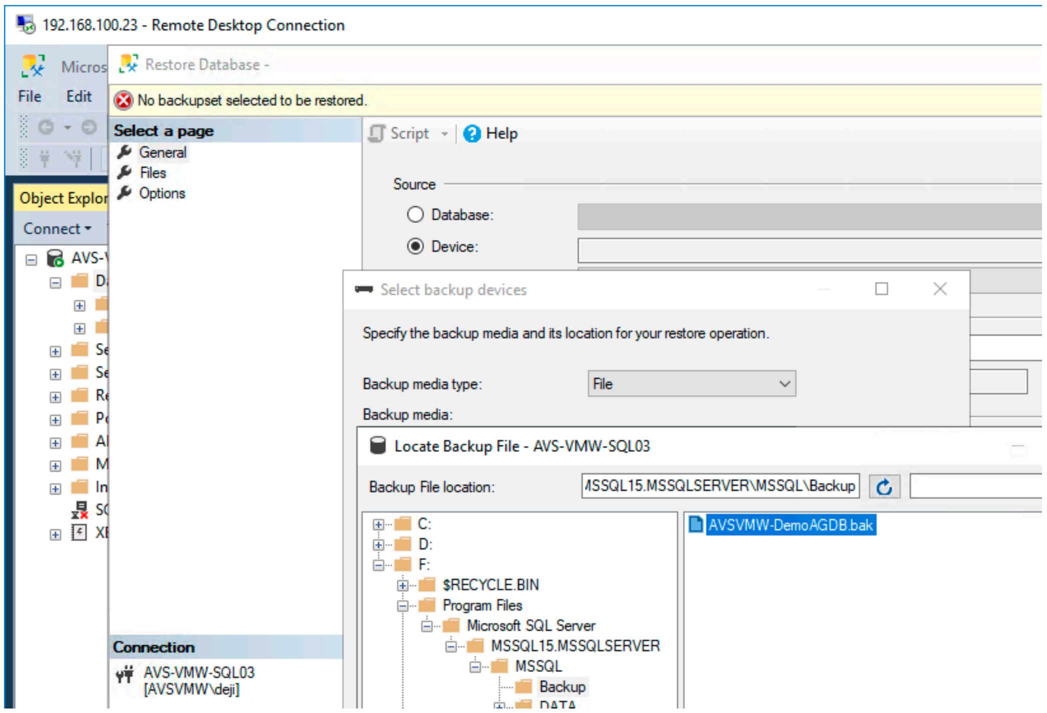
• Next, we restore the backed-up database to SQL03. This helps us avoid lengthy re-seed and synchronization, especially for large and highly active Databases


192.168.100.23 - Remote Desktop Connection

Microsoft SQL Server Management Studio

AVS-VMW-SQL03 (SQL Server 15.0.2000.5 - AVSVMW\deji)

- We select our .bak file.


192.168.100.23 - Remote Desktop Connection

Restore Database -

No backupset selected to be restored.

Select a page: General, Files, Options

Source
- Database:
- Device: [✓]

Select backup devices

Specify the backup media and its location for your restore operation.

Backup media type: File

Backup media:

Locate Backup File - AVS-VMW-SQL03

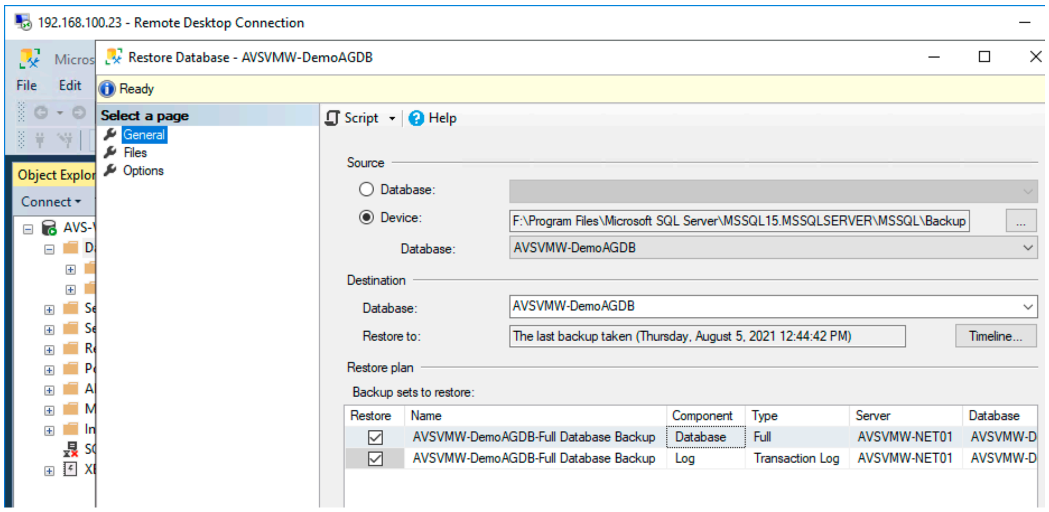
Backup File location: \SSQL15.MSSQLSERVER\MSSQL\Backup

```
C:
D:
F:
  $RECYCLE.BIN
  Program Files
    Microsoft SQL Server
      MSSQL15.MSSQLSERVER
        MSSQL
          Backup
          DATA
```

AVSVMW-DemoAGDB.bak

Connection: AVS-VMW-SQL03 [AVSVMW\deji]

- We verify that all looks good.


192.168.100.23 - Remote Desktop Connection

Restore Database - AVSVMW-DemoAGDB

Ready

Select a page: General, Files, Options

Source
- Database:
- Device: [✓] F:\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\Backup
- Database: AVSVMW-DemoAGDB

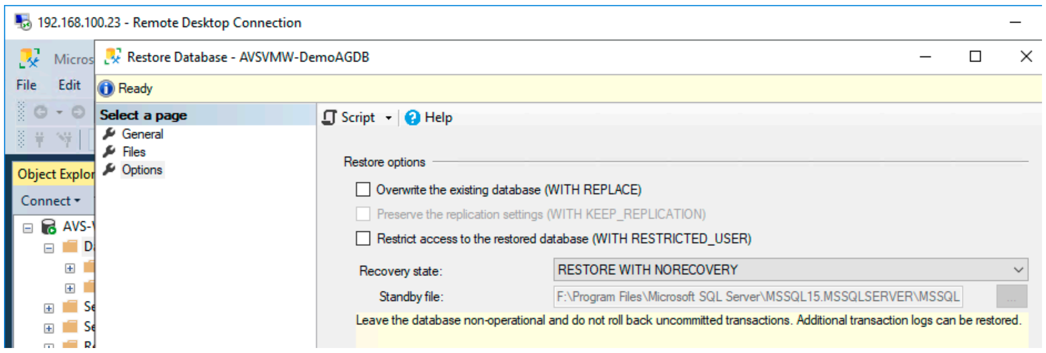
Destination
- Database: AVSVMW-DemoAGDB
- Restore to: The last backup taken (Thursday, August 5, 2021 12:44:42 PM)

Restore plan

Backup sets to restore:

Restore	Name	Component	Type	Server	Database
✓	AVSVMW-DemoAGDB-Full Database Backup	Database	Full	AVSVMW-NET01	AVSVMW-D
✓	AVSVMW-DemoAGDB-Full Database Backup	Log	Transaction Log	AVSVMW-NET01	AVSVMW-D

- We choose the **Restore with NoRecovery** option and *OK* the choices.


192.168.100.23 - Remote Desktop Connection
Restore Database - AVSVMW-DemoAGDB
Ready
Select a page
General
Files
Options
Script
Help

Restore options

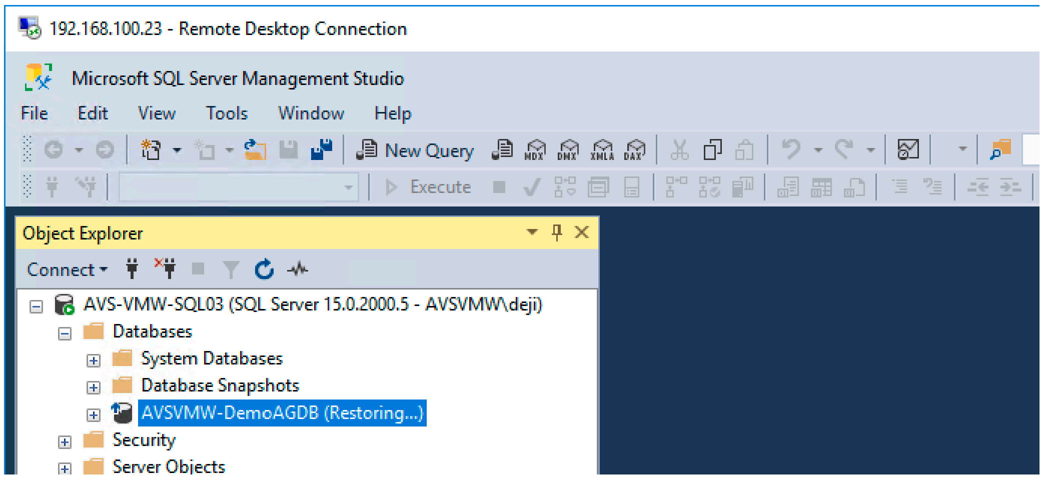
- [ ] Overwrite the existing database (WITH REPLACE)
- [ ] Preserve the replication settings (WITH KEEP_REPLICATION)
- [ ] Restrict access to the restored database (WITH RESTRICTED_USER)

Recovery state: RESTORE WITH NORECOVERY

Standby file: F:\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL

Leave the database non-operational and do not roll back uncommitted transactions. Additional transaction logs can be restored.

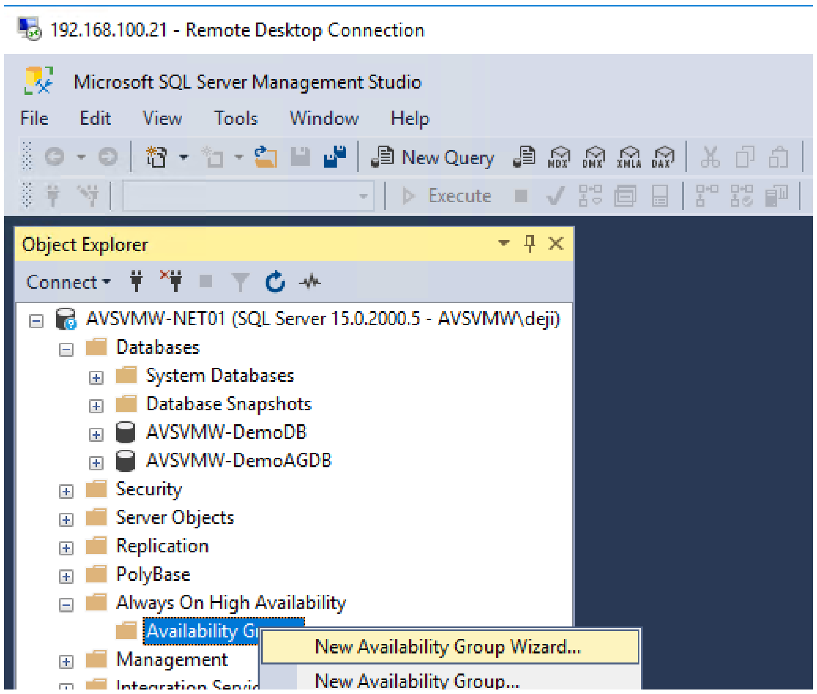
- Depending on the size of the Database, the restore process may take some time to complete.


192.168.100.23 - Remote Desktop Connection
Microsoft SQL Server Management Studio
File
Edit
View
Tools
Window
Help
New Query
Execute
Object Explorer
Connect

AVS-VMW-SQL03 (SQL Server 15.0.2000.5 - AVSVMW\deji)
Databases
System Databases
Database Snapshots
AVSVMW-DemoAGDB (Restoring...)
Security
Server Objects

- Now that we have a stub copy of the Database on the second node, we are ready to configure it for high availability, using the Microsoft SQL Server Availability Group option.

- From the **Always On Availability Group** menu on the Active node (SQL01), select **New Availability Group Wizard**.

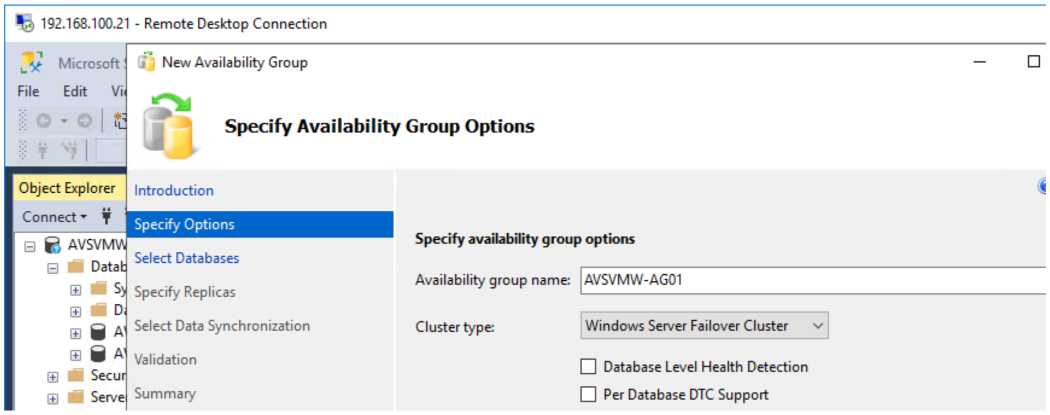


192.168.100.21 - Remote Desktop Connection

Microsoft SQL Server Management Studio

AVSVMW-NET01 (SQL Server 15.0.2000.5 - AVSVMW\deji)
Databases
System Databases
Database Snapshots
AVSVMW-DemoDB
AVSVMW-DemoAGDB
Security
Server Objects
Replication
PolyBase
Always On High Availability
Availability Gr
Management
Integration Servic

- We give the AG a name.



192.168.100.21 - Remote Desktop Connection

New Availability Group

Specify Availability Group Options

Specify availability group options

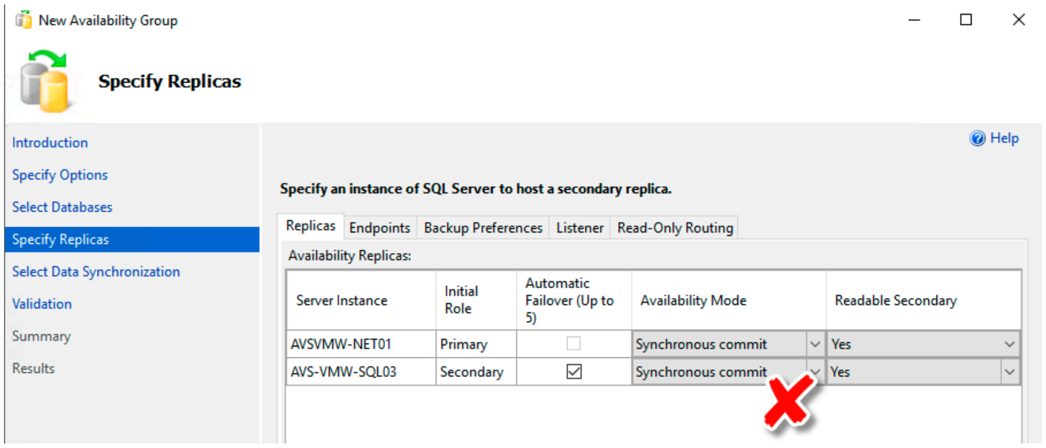
Availability group name: AVSVMW-AG01

Cluster type: Windows Server Failover Cluster

☐ Database Level Health Detection

☐ Per Database DTC Support

- We want our Replica copy of the AG to be in Asynchronous mode, so be sure to toggle the **Availability Mode** from **Synchronous Commit** to **Asynchronous Commit**.

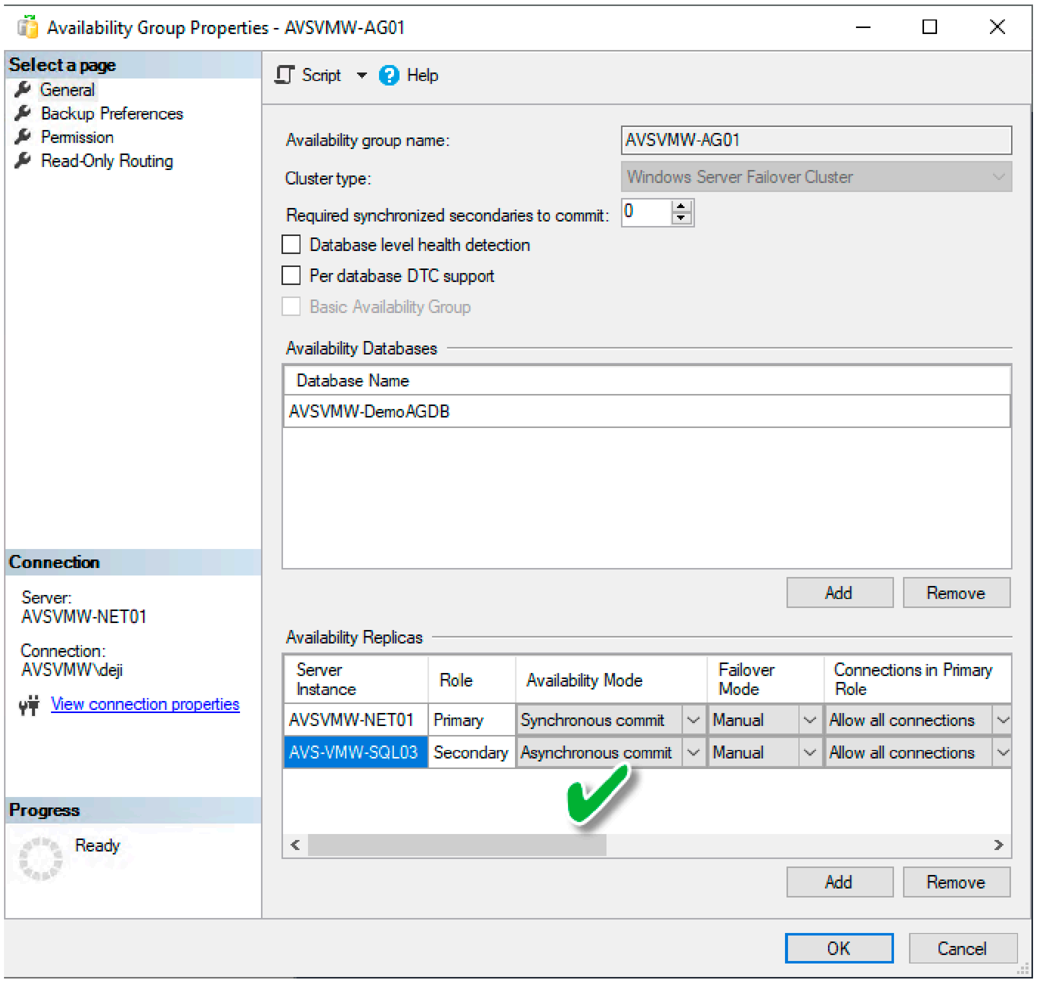


Specify an instance of SQL Server to host a secondary replica.

Availability Replicas:

Server Instance	Initial Role	Automatic Failover (Up to 5)	Availability Mode	Readable Secondary
AVSVMW-NET01	Primary	☐	Synchronous commit	Yes
AVS-VMW-SQL03	Secondary	☑	Synchronous commit	Yes

- Here is what we want our **Availability Mode** to look like for our AG Database configuration.



Server:
AVSVMW-NET01

Connection:
AVSVMW\deji

Availability group name: AVSVMW-AG01

Cluster type: Windows Server Failover Cluster

Required synchronized secondaries to commit: 0

☐ Database level health detection

☐ Per database DTC support

☐ Basic Availability Group

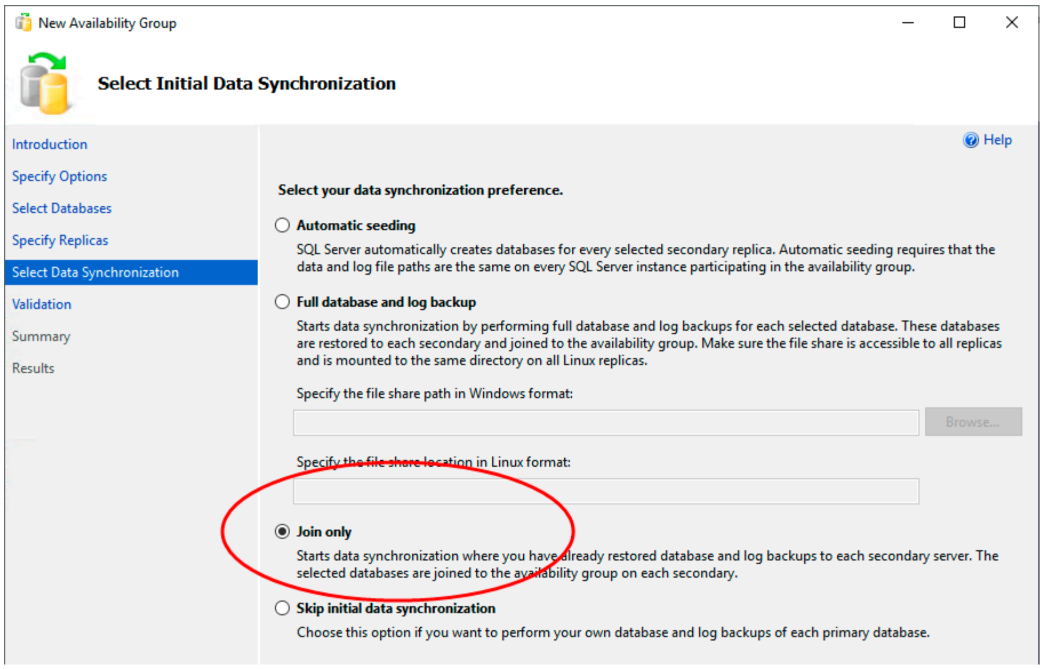
Availability Databases

Database Name
AVSVMW-DemoAGDB

Availability Replicas

Server Instance	Role	Availability Mode	Failover Mode	Connections in Primary Role
AVSVMW-NET01	Primary	Synchronous commit	Manual	Allow all connections
AVS-VMW-SQL03	Secondary	Asynchronous commit	Manual	Allow all connections

- Choose the **Join Only** option – remember, we have already copied the Database manually to the Secondary node.



New Availability Group

Select Initial Data Synchronization

Introduction
Specify Options
Select Databases
Specify Replicas
Select Data Synchronization
Validation
Summary
Results

Help

Select your data synchronization preference.

Automatic seeding
SQL Server automatically creates databases for every selected secondary replica. Automatic seeding requires that the data and log file paths are the same on every SQL Server instance participating in the availability group.

Full database and log backup
Starts data synchronization by performing full database and log backups for each selected database. These databases are restored to each secondary and joined to the availability group. Make sure the file share is accessible to all replicas and is mounted to the same directory on all Linux replicas.

Specify the file share path in Windows format:

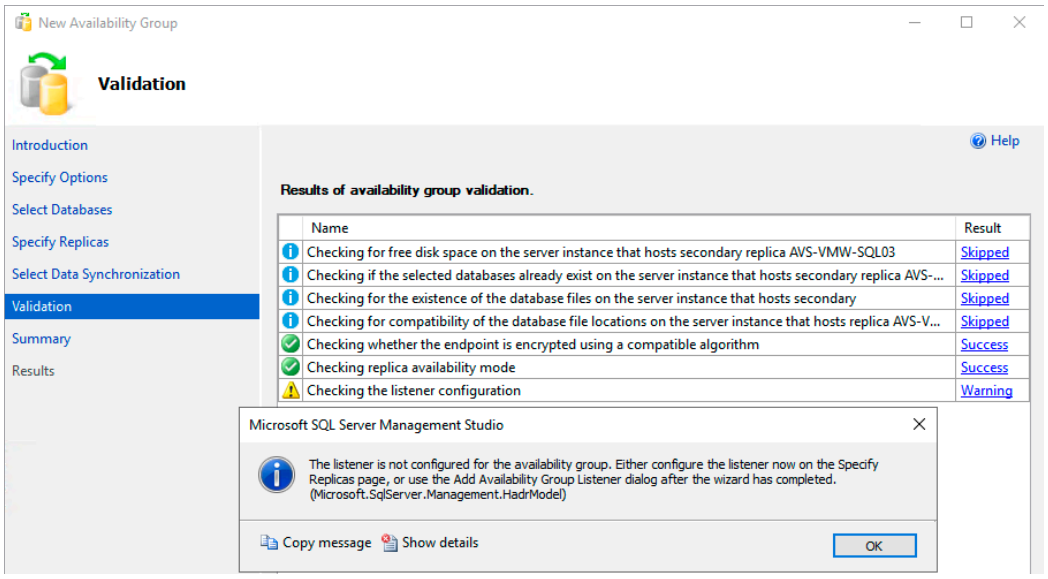
Browse...

Specify the file share location in Linux format:

[✓ **Join only**]
Starts data synchronization where you have already restored database and log backups to each secondary server. The selected databases are joined to the availability group on each secondary.

Skip initial data synchronization
Choose this option if you want to perform your own database and log backups of each primary database.

- We skipped the option to configure the Listener for now. We will come back to that later.



New Availability Group

Validation

Introduction
Specify Options
Select Databases
Specify Replicas
Select Data Synchronization
Validation
Summary
Results

Help

Results of availability group validation.

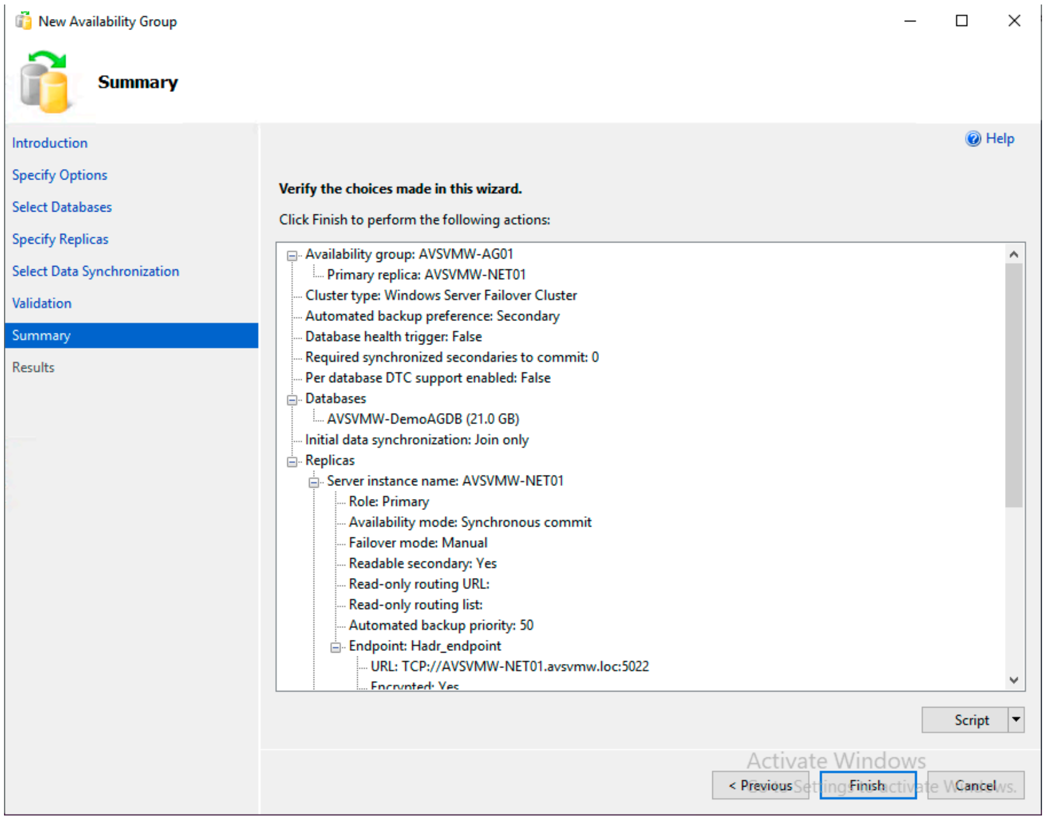
	Name	Result
ℹ	Checking for free disk space on the server instance that hosts secondary replica AVS-VMW-SQL03	Skipped
ℹ	Checking if the selected databases already exist on the server instance that hosts secondary replica AVS-...	Skipped
ℹ	Checking for the existence of the database files on the server instance that hosts secondary	Skipped
ℹ	Checking for compatibility of the database file locations on the server instance that hosts replica AVS-V...	Skipped
✓	Checking whether the endpoint is encrypted using a compatible algorithm	Success
✓	Checking replica availability mode	Success
⚠	Checking the listener configuration	Warning

Microsoft SQL Server Management Studio

The listener is not configured for the availability group. Either configure the listener now on the Specify Replicas page, or use the Add Availability Group Listener dialog after the wizard has completed. (Microsoft.SqlServer.Management.HadrModel)

Copy message   Show details   OK

- We're almost done. Let's click **Finish**.



New Availability Group

Summary

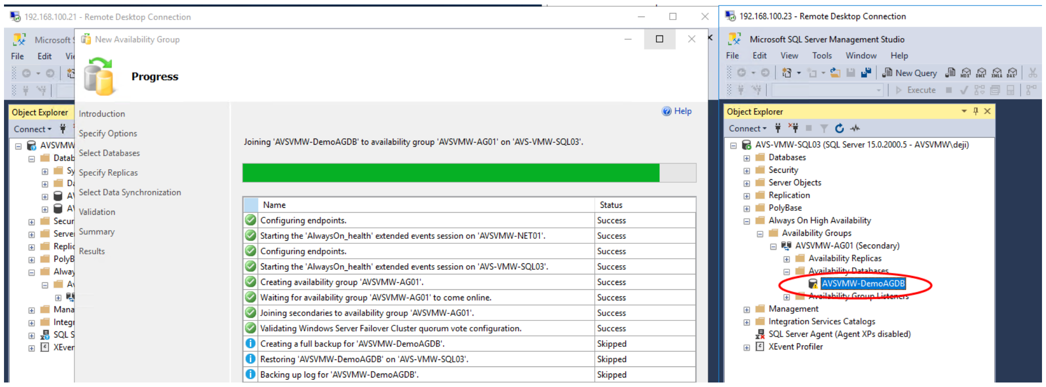
Introduction
Specify Options
Select Databases
Specify Replicas
Select Data Synchronization
Validation
Summary
Results

Help

Verify the choices made in this wizard.

Click Finish to perform the following actions:

```
Availability group: AVSVMW-AG01
    Primary replica: AVSVMW-NET01
Cluster type: Windows Server Failover Cluster
Automated backup preference: Secondary
Database health trigger: False
Required synchronized secondaries to commit: 0
Per database DTC support enabled: False
Databases
    AVSVMW-DemoAGDB (21.0 GB)
Initial data synchronization: Join only
Replicas
    Server instance name: AVSVMW-NET01
        Role: Primary
        Availability mode: Synchronous commit
        Failover mode: Manual
        Readable secondary: Yes
        Read-only routing URL:
        Read-only routing list:
        Automated backup priority: 50
        Endpoint: Hadr_endpoint
            URL: TCP://AVSVMW-NET01.avsvmw.loc:5022
            Encrypted: Yes
```



192.168.100.21 - Remote Desktop Connection

New Availability Group

Progress

Introduction
Specify Options
Select Databases
Specify Replicas
Select Data Synchronization
Validation
Summary
Results

Joining 'AVSVMW-DemoAGDB' to availability group 'AVSVMW-AG01' on 'AVS-VMW-SQL03'.

Name	Status
Configuring endpoints.	Success
Starting the 'AlwaysOn_health' extended events session on 'AVSVMW-NET01'.	Success
Configuring endpoints.	Success
Starting the 'AlwaysOn_health' extended events session on 'AVS-VMW-SQL03'.	Success
Creating availability group 'AVSVMW-AG01'.	Success
Waiting for availability group 'AVSVMW-AG01' to come online.	Success
Joining secondaries to availability group 'AVSVMW-AG01'.	Success
Validating Windows Server Failover Cluster quorum vote configuration.	Success
Creating a full backup for 'AVSVMW-DemoAGDB'.	Skipped
Restoring 'AVSVMW-DemoAGDB' on 'AVS-VMW-SQL03'.	Skipped
Backing up log for 'AVSVMW-DemoAGDB'.	Skipped

192.168.100.23 - Remote Desktop Connection

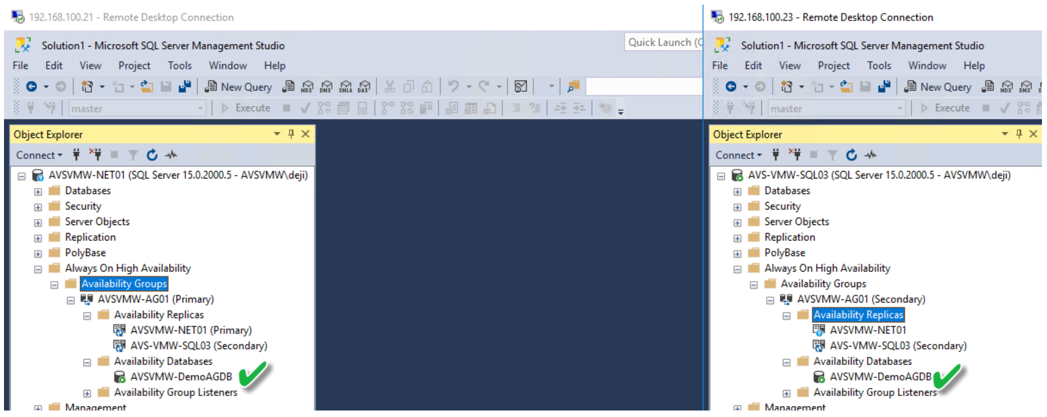
Microsoft SQL Server Management Studio

Object Explorer

```
AVS-VMW-SQL03 (SQL Server 15.0.2000.5 - AVSVMW\deji)
    Databases
    Security
    Server Objects
    Replication
    PolyBase
    Always On High Availability
        Availability Groups
            AVSVMW-AG01 (Secondary)
                Availability Replicas
                Availability Databases
                    AVSVMW-DemoAGDB
                Availability Group Listeners
    Management
    Integration Services Catalogs
    SQL Server Agent (Agent XPs disabled)
    XEvent Profiler
```

- The wizard skips several of the steps because we've already completed them in previous step.
- Note the state of the Database on SQL03. We're cooking with gas now.

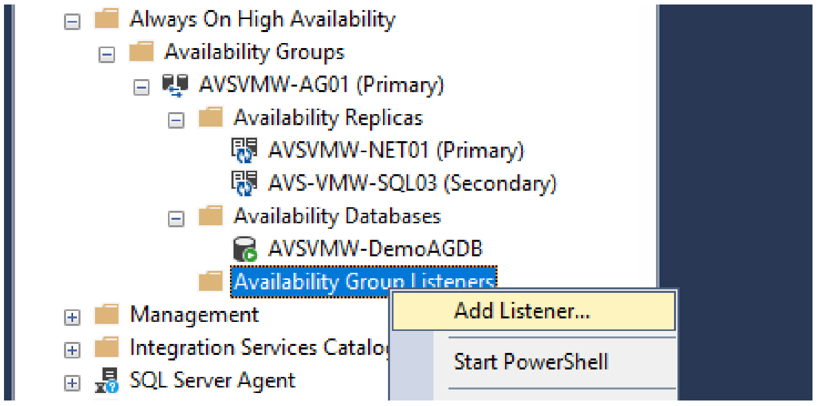
- The process takes a while, but, eventually, here is the result of our effort.

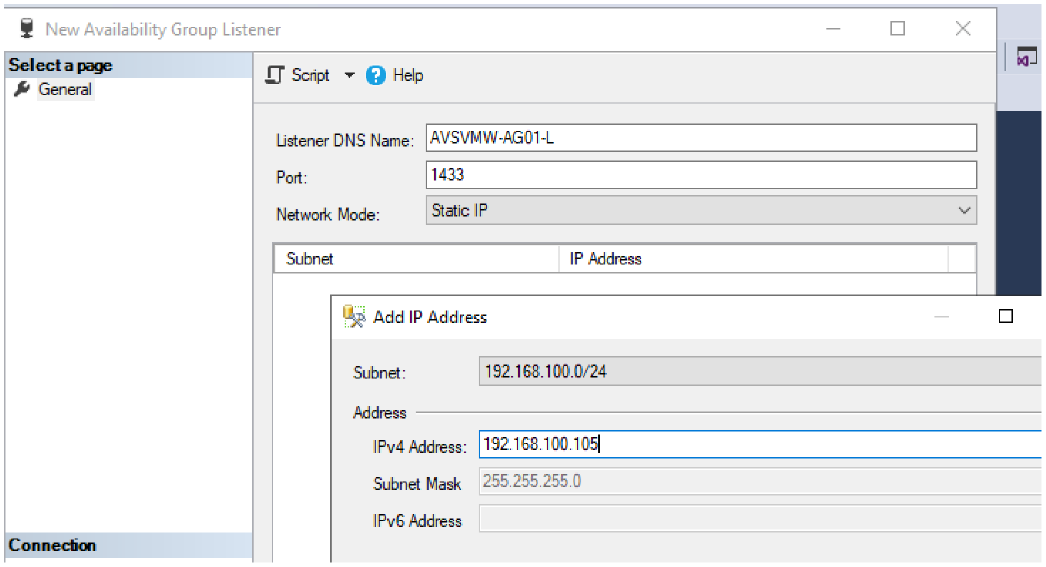

192.168.100.21 - Remote Desktop Connection
Solution1 - Microsoft SQL Server Management Studio
Object Explorer
AVSVMW-NET01 (SQL Server 15.0.2000.5 - AVSVMW\deji)
Databases
Security
Server Objects
Replication
PolyBase
Always On High Availability
Availability Groups
AVSVMW-AG01 (Primary)
Availability Replicas
AVSVMW-NET01 (Primary)
AVS-VMW-SQL03 (Secondary)
Availability Databases
AVSVMW-DemoAGDB
Availability Group Listeners
Management
192.168.100.23 - Remote Desktop Connection
Solution1 - Microsoft SQL Server Management Studio
Object Explorer
AVS-VMW-SQL03 (SQL Server 15.0.2000.5 - AVSVMW\deji)
Databases
Security
Server Objects
Replication
PolyBase
Always On High Availability
Availability Groups
AVSVMW-AG01 (Secondary)
Availability Replicas
AVSVMW-NET01
AVS-VMW-SQL03 (Secondary)
Availability Databases
AVSVMW-DemoAGDB
Availability Group Listeners
Management

Now, we are ready to configure our Listener.

– **NOTE:** In our simplified configuration, we have a stretched/extended/flat network segment between our on-premises SQL Servers and AVS. We're using the same subnet for both sides. In this configuration, we need only one Listener. If this were routed or disjointed segments, we would have needed to create a Listener for each segment where we have the clustered VMs.

- In SQL Server Manager, expand **Always On High Availability** and navigate to the **Availability Group Listeners** section. Right click and select **Add Listener**.

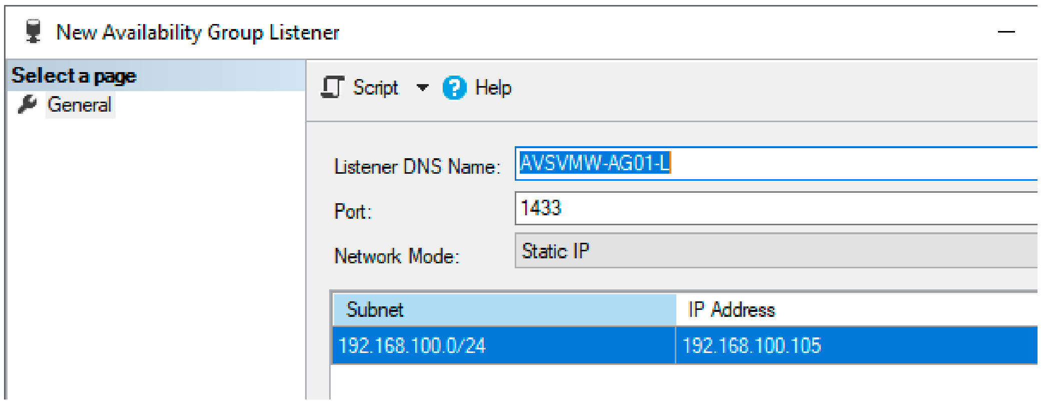




Listener DNS Name: AVSVMW-AG01-L
Port: 1433
Network Mode: Static IP

Subnet: 192.168.100.0/24
IPv4 Address: 192.168.100.105
Subnet Mask 255.255.255.0
IPv6 Address

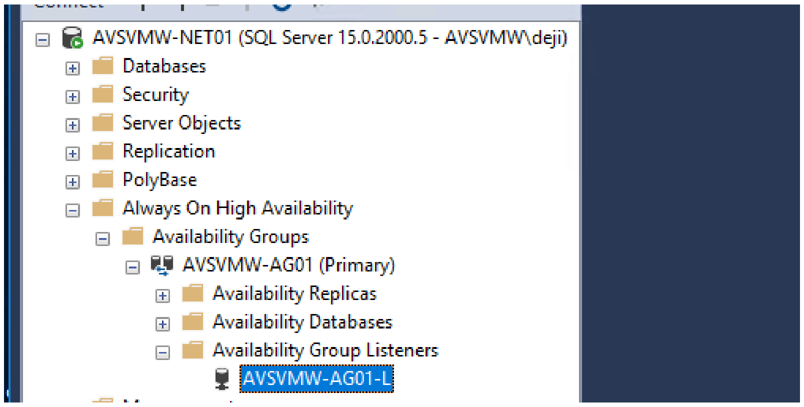
- Give the Listener a name and an IP Address.
  - As you can see, the Listener has a **DNS Name**. This is part of the administrative pre-tasks not covered in this document because it is not peculiar to virtualization.
  - Also note that we left the port at the default **1433**.



Listener DNS Name: AVSVMW-AG01-L
Port: 1433
Network Mode: Static IP

Subnet	IP Address
192.168.100.0/24	192.168.100.105

- Click **Next** and let's wrap up the setup.

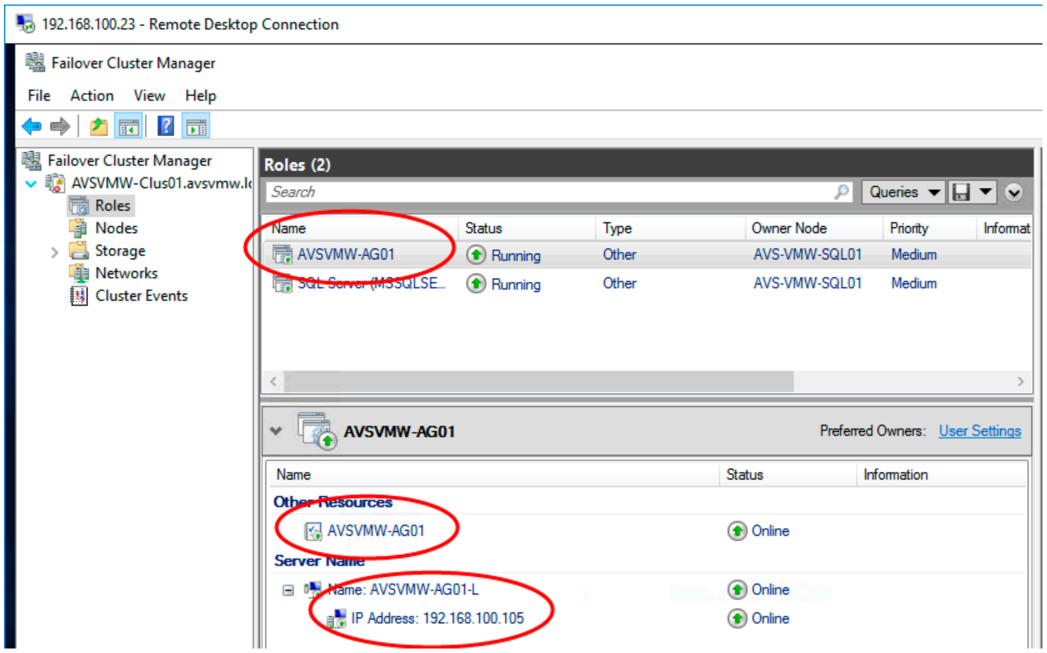


AVSVMW-NET01 (SQL Server 15.0.2000.5 - AVSVMW\deji)
- Databases
- Security
- Server Objects
- Replication
- PolyBase
- Always On High Availability
  - Availability Groups
    - AVSVMW-AG01 (Primary)
      - Availability Replicas
      - Availability Databases
      - Availability Group Listeners
        - AVSVMW-AG01-L

• There it is.

If we look in Windows Cluster Manager, we can now see that we have two clustered Microsoft SQL Server resources (Roles).

One is our newly clustered, share-nothing Availability Group Database



192.168.100.23 - Remote Desktop Connection

Failover Cluster Manager

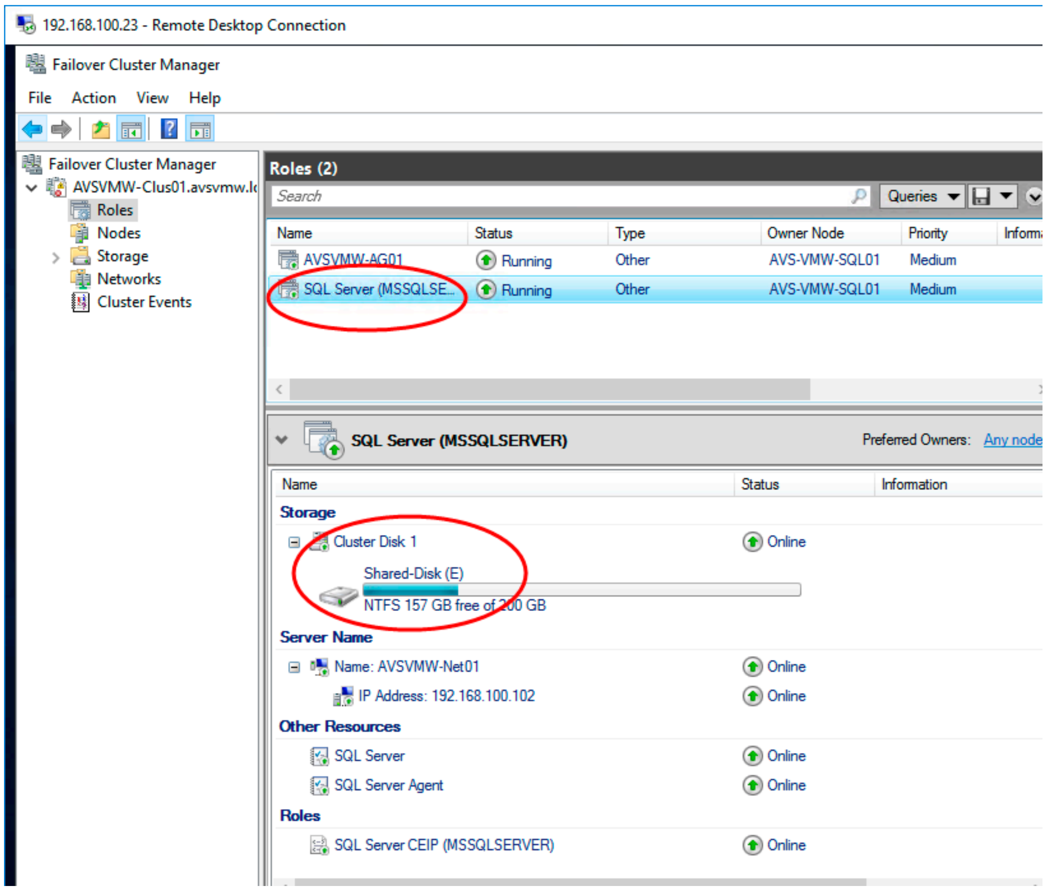
Roles (2)

Name	Status	Type	Owner Node	Priority
AVSVMW-AG01	Running	Other	AVS-VMW-SQL01	Medium
SQL Server (MSSQLSE...	Running	Other	AVS-VMW-SQL01	Medium

AVSVMW-AG01 — Preferred Owners: User Settings

Name	Status	Information
Other Resources		
AVSVMW-AG01	Online	
Server Name		
Name: AVSVMW-AG01-L	Online	
IP Address: 192.168.100.105	Online	

And the other is the shared-disk Failover Cluster Instance.



Name	Status	Type	Owner Node	Priority	Inform
AVSVMW-AG01	Running	Other	AVS-VMW-SQL01	Medium	
SQL Server (MSSQLSE...	Running	Other	AVS-VMW-SQL01	Medium	

SQL Server (MSSQLSERVER) Preferred Owners: Any node

Name	Status	Information
Storage		
Cluster Disk 1	Online	
Shared-Disk (E) NTFS 157 GB free of 200 GB		
Server Name		
Name: AVSVMW-Net01	Online	
IP Address: 192.168.100.102	Online	
Other Resources		
SQL Server	Online	
SQL Server Agent	Online	
Roles		
SQL Server CEIP (MSSQLSERVER)	Online	

Conclusion

Mission accomplished. We have successfully configured a functioning, production-ready, highly-available Microsoft SQL Server farm on our Azure VMware Solution platforms, using a mixture of the two most common application-level HA options.

At its core, Azure VMware Solution is a VMware vSphere-base hybrid cloud platform, allowing administrators to seamlessly and easily extend their on-premises infrastructure to the public cloud, using the familiar, true-and-tested vSphere technologies they are already familiar and comfortable with, without unnecessarily re-factoring their applications or re-learning complex processes and ways of doing things.

As you no doubt have noticed throughout the entire exercise described in this document, there is hardly any difference between performing these tasks in your *regular* vSphere infrastructure and doing it on AVS. This is one of the core value propositions of the Azure VMware Solution: Do more, with same tools, skills and experience, and without a steep learning curve.