IDERA® | SQL Server Tools Whitepaper

TOP FIVE SQL SERVER CLUSTER SETUP MISTAKES

BY KENDRA LITTLE

TABLE OF CONTENTS

SOLUTION OVERVIEW

Windows Failover Clustering can keep your SQL Server databases online and your customers happy when hardware or operating system problems strike. Failover Clustering also makes your life easier when you need to do maintenance on a physical server: just fail your SQL Server instance over to another server and complete your work with minimal downtime.

However, be careful: If you misconfigure your cluster, then you are bound for poor SQL Server performance or extended unplanned downtime.

You should not let an otherwise great high availability feature cause weak performance or outages. Let us learn how to avoid some of the most prevalent SQL cluster setup mistakes.

Need a quick refresher on precisely what a failover cluster is? Read An Introduction to SQL Server Clusters to get the basics.

MISTAKE #5: NOT PLANNING YOUR NODES AND INSTANCES

It is essential that you properly plan your nodes and instances. Start by considering the following questions:

HOW MANY CLUSTER NODES WILL YOU BUILD?

Nodes are the servers configured as members of the Windows Failover Cluster. Each node may be part of only one Failover Cluster and must be a member of a Windows domain.

Most people plan Windows Failover Clusters for SQL Servers, where both performance and availability are important. If this is the case, plan one node per SQL Server instance plus at least one additional passive node.

Ask yourself, “How many servers can fail before performance is degraded?” That answer is the number of passive nodes you need in your cluster.

HOW MANY SQL SERVER INSTANCES WILL YOU INSTALL?

Each SQL Server instance can move from node to node, but each instance can be active on only one node at any given time. Every SQL Server instance has its collation, user and system databases, logins, and SQL Agent jobs configured, and all those components move along with the instance.

HOW MANY WINDOWS FAILOVER CLUSTERS WILL YOU MANAGE?

You might be tempted to put all your nodes and SQL Server instances in a single cluster. That seems attractive for cost savings on hardware: a four-node cluster with one passive node sounds better than a pair of two-node clusters, each with a passive node. However, consider the following factors:

- SQL Server Licensing Costs: SQL Server Standard Edition allows you to use two nodes for a single instance in a Windows Failover Cluster. To install the instance on three or more nodes, you need SQL Server Enterprise Edition, which costs significantly more. Software Assurance may be required for the passive node. (i)
- Uptime Requirements: You should apply hotfixes and service packs to Windows and SQL Server regularly. After applying updates, you should test that each SQL Server instance runs successfully on every node. The more nodes and the more instances you have in a cluster, the more downtime each instance will have when you fail it over between nodes.

WHAT KIND OF STORAGE WILL YOU USE?

For a Windows Failover Cluster Instance, all nodes must have access to shared storage: each SQL Server user and system database anywhere in the cluster will have just one copy of its data stored in a central location. (Foreshadowing: there is one exception to this. Read on!) The shared storage may be a large, sophisticated storage area network, or a smaller storage array.

WHAT WILL YOUR STAGING ENVIRONMENT BE LIKE?

To support clustered SQL Server in production, you need a non-production SQL Server staging cluster. The staging cluster should match production as closely as possible.

SOME FREQUENTLY ASKED QUESTIONS ABOUT DESIGNING CLUSTERS FOR SQL SERVER:

- Default SQL Server Instances: You may have only one Default SQL Server instance per Windows Failover Cluster. Do not worry too much about that limitation: you may set all instances on a cluster to use port 1433 because each instance will have its IP address.
- Stretch Clusters in Multiple Datacenters: Your Windows Failover Cluster can span subnets and data centers, but you need to have storage replication manage the data transfer (ii). If you would like SQL Server to manage the data transfer instead of storage replication, investigate SQL Server Availability Groups. Availability Group instances do not require shared storage and may be combined with Windows Failover Cluster instances. If you combine the technologies, you will only have automatic failover within the Windows Failover Cluster Instance. (iii)

MISTAKE #4: BUYING THE WRONG HARDWARE

It is easy to get confused when sizing hardware for a clustered SQL Server instance. There are tons of misinformation out there.

“BUY ENOUGH MEMORY. HOWEVER, DO NOT ASSUME YOU ARE LIMITED TO USING HALF OF IT!”

Do not go cheap on memory: it is critical to SQL Server performance.

Many people incorrectly believe that if they have two instances on a two-node cluster, they need to set SQL Server’s Max Server Memory setting to 50% or less of the total memory of each node to protect performance in case both instances need to run on a single node.

Do not do that. On SQL Server 2005, if multiple SQL Server instances share a single node, they will negotiate with each other and balance memory use. You may use the Min Server Memory setting to prioritize one instance over another. (v)

This approach is also applicable when there is another memory-intensive process on the computer since it would ensure that SQL Server would at least get a reasonable amount of memory.

“BUY MULTIPLE PHYSICAL NETWORK INTERFACE CONTROLLERS.”

As of Windows Server 2008 and higher, you no longer need to configure a separate heartbeat network for a Windows Failover cluster. Instead, you must make sure that your network path is redundant.

That means you need two physical network adapters or network interface controllers in each cluster node. You may bring these together into a single network team; the point is that communication must be able to continue if either physical network interface controller fails.

The network interface controllers must be connected to a network that has redundant switches, and this network is only for your SQL Server and cluster traffic. In other words, your storage network should not use this. If you use Internet small computer systems interface storage, it must have its own dedicated network interface controllers. (vi)

“BUY THE RIGHT TEMPDB STORAGE.”

There is an exception to the rule mentioned above that all user and system databases must be on shared storage. Beginning in SQL Server 2012, you can configure tempdb on local storage for a clustered SQL Server instance. When performance is important, speeding up tempdb is attractive. solid state drive can be expensive to implement in shared storage devices.

Lots of people choose to install solid state drive into each server node to make tempdb fast. The solid state drives may be either 2.5-inch solid state drives or peripheral component interconnect Express cards. If you take this approach, remember:

- All nodes need identical solid state drive configurations. When your SQL Server instance fails over and tries to come upon another node, it needs to find tempdb at the right drive letter with the right available capacity.
- Do not buy just one solid state drive per node. Failure of a drive under tempdb can be disastrous for a SQL Server instance. Even though you are using a Windows Failover Cluster, you should provision hardware to avoid unplanned failovers as much as possible. If any transactions are in-flight when a failure occurs, the process of bringing databases online on a new node may take longer than your users like.
- Your staging environment needs the same solid state drive configuration. Imagine that you have started seeing a periodic slow performance from your solid state drives. A firmware upgrade is available from the vendor. Where do you want to test the firmware upgrade first?

MISTAKE #3: SELECTING THE WRONG VERSION OF WINDOWS OR SQL SERVER

When you are planning a new Windows Failover Cluster, the Windows Server version is important. Selecting the right Windows installation can make your cluster easier to manage and easier to keep online.

- Windows Server 2008 introduced more robust networking capabilities and critical tools, including the Cluster Validation Wizard.
- Windows Server 2012 brought a major improvement: the cluster determines cluster majority by looking at the current active members of the cluster (not the number of members that were initially configured). That helps keep your SQL Server instance online when failures start to occur.
- Windows Server 2012 R2 added better cluster options to manage witness votes and tie situations.
- Windows Server 2016 has two important enhancements with coordinated universal time and local time, making it easier to deal with multiple time zones.
- Windows Server 2019 has new enhancement about cluster hardening where the requirement of the active directory is eliminated for clusters in Windows Server

However, do not base your decision only on all the improved features.

It is crucial to determine the best patch configuration for your windows server and select the latest version of SQL Server if possible.

USE THE LATEST WINDOWS SERVER WHENEVER POSSIBLE

If you are planning a new cluster for your SQL Server, identify the highest Windows Server version supported by your company. Always map out all the improvements the new operating system brings and the hotfixes it helps you avoid: your company may not support a new OS until you advocate for change.

DO NOT MIX SQL SERVER VERSIONS

Sometimes people ask if they can install different SQL Server versions on a single Windows Failover cluster. That is an uncommon configuration. It is called a side by side cluster deployment.

Although this is technically possible, I do not recommend it. Installation is complex: you must install each SQL Server version from lowest to highest when setting up every node.

Even worse, whenever you need to troubleshoot or seek support for an incident, your very specialized configuration makes it more di-cult to resolve and find the root cause for the problem.

MISTAKE #2: CONFIGURING QUORUM BADLY

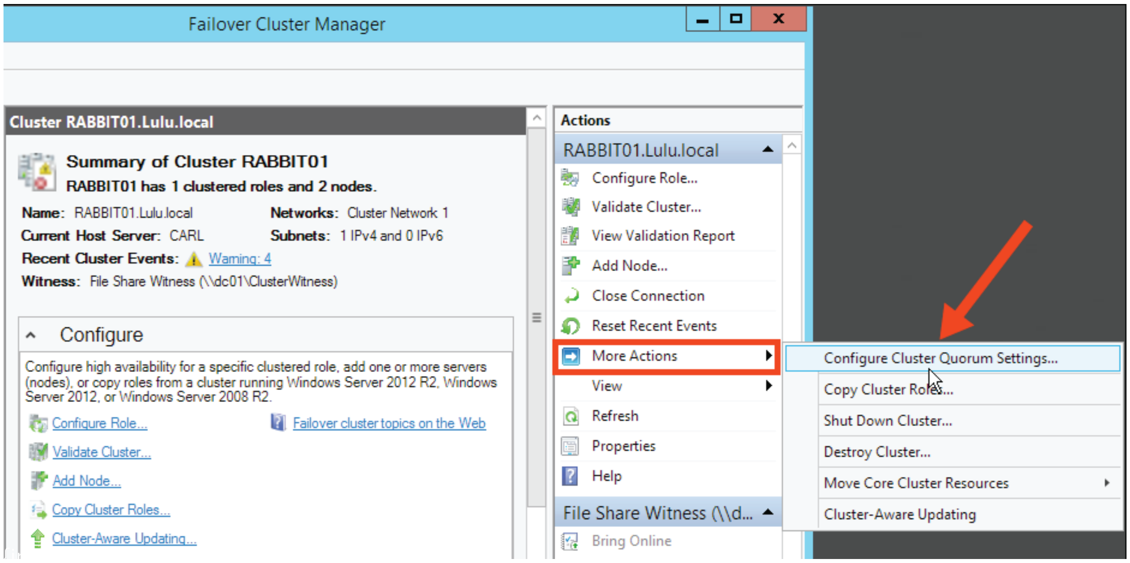
You must understand quorum and configure it properly in the Windows Failover Cluster to keep your SQL Server databases online. That is tricky. The best practices for configuring quorum vary depending on the version of Windows you are using, the number of nodes you have, and the reliability of network communication between the nodes.

Failover clusters want to keep your databases online, but they have a problem: they must keep each SQL Server instance online only on a single node!

Quorum is a process by which each member of the cluster stops and takes attendance and checks which members of the cluster own each resource in the cluster. Failover Clustering got a lot better at checking quorum in Windows Server 2012 and the latest versions: instead of always counting the initial cluster configuration as the total number of possible votes, at any given time, the current members of the cluster are counted as the voting population.

Potential voting members of the cluster are:

- Each node of the cluster
- A witness (optional): This is a disk resource (or a file share) that keeps a copy of critical information about the cluster's state and configuration



Failover Cluster Manager

Cluster RABBIT01.Lulu.local

Summary of Cluster RABBIT01

RABBIT01 has 1 clustered roles and 2 nodes.

Name: RABBIT01.Lulu.local
Networks: Cluster Network 1
Current Host Server: CARL
Subnets: 1 IPv4 and 0 IPv6
Recent Cluster Events: Warning: 4
Witness: File Share Witness (\\dc01\ClusterWitness)

Configure

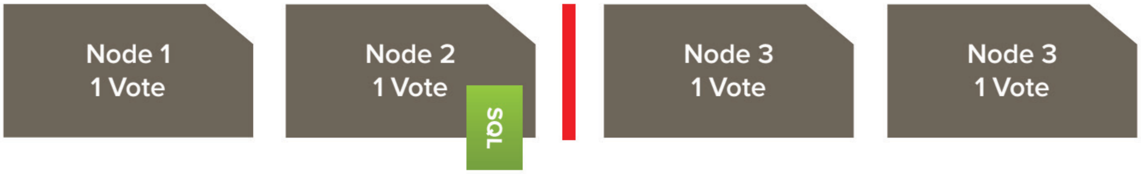
Configure high availability for a specific clustered role, add one or more servers (nodes), or copy roles from a cluster running Windows Server 2012 R2, Windows Server 2012, or Windows Server 2008 R2.

The Golden Rule: Every time your cluster configuration changes, re-evaluate quorum to avoid a tie situation.

Your cluster always wants to have an odd number of votes available. Clusters just hate a tie because it is challenging for them to know what to do!

Imagine you have a Windows Failover cluster with four nodes. You have not configured a disk witness of any kind. Each node is granted a single vote, so you have a total of four votes. One SQL Server instance is installed on the cluster, and it is currently residing on Node 2.

Suddenly, a network crisis occurs! Nodes 1 and 2 can see one another, but not Nodes 3 and 4. Nodes 3 and 4 can see each other, but neither can see Nodes 1 and 2.



What happens now when each node counts votes? Node 1 and Node 2 can see each other and count two votes out of four: that is not a majority. Node 3 and Node 4 are in the same predicament. Neither group has a majority: that means neither group has the unequivocal right to keep the SQL Server instance online.

Depending on your SQL Server version and whether you have established a Tiebreaker (available in SQL Server 2012R2 and onwards), the Windows Failover Cluster service may shut down on every node and take your databases offline with it.

Good news: you can decrease the chances of this happening!

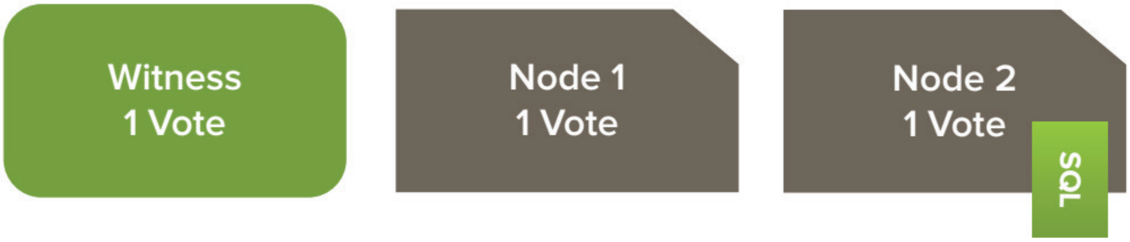
ASSIGN A WITNESS TO GET AN ODD NUMBER OF VOTES

Every cluster may have one witness. It also might not have a witness: it is up to you to decide.

- The witness is a member of the cluster that helps the cluster decide where your SQL Server Instances should be active. You may choose to set up a witness using one of these configurations:
  A Disk Witness (This is called Quorum Disk sometimes)
- A File Share Witness

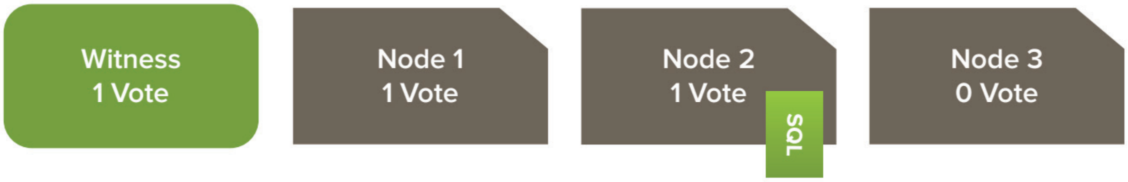
The Disk Witness may be a little more work to configure because this disk must be available on shared storage available to all of the nodes. However, the disk witness is a bit more sophisticated than the file share witness and holds more information that may be useful to the cluster when failures occur.

If you decide to use a witness, use a Disk Witness whenever possible.
Here we have got a two-node Failover Cluster. To make sure there is an odd number of votes, the administrator configured quorum to use a witness:



REMOVE VOTES WHEN APPROPRIATE

You can manually remove the vote of a node. Removing a vote is useful, particularly when you have a cluster that spans data centers. In this example, Node 1 and Node 2 are in a primary datacenter. Node 3 is in a remote datacenter, and network communication is unreliable. You will only ever manually failover to the remote datacenter.

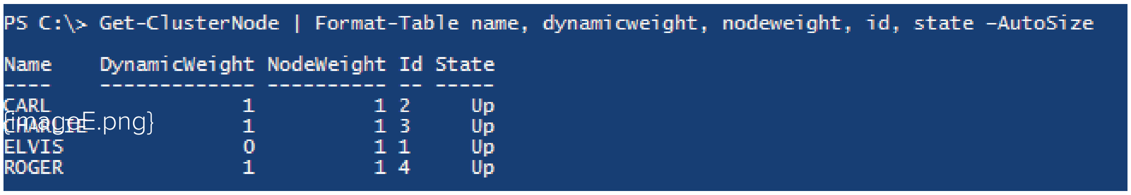


In this scenario, you can manually remove the vote from the node in the remote data center and use a witness to achieve an odd number of votes in the primary datacenter. Windows Server 2012 and onwards supports Dynamic Quorum features.

WINDOWS SERVER DYNAMIC QUORUM DYNAMICALLY ADJUSTS NODE VOTES

The Dynamic Quorum feature makes life easier. When this is enabled, the cluster will count up current members of the cluster and decide if it should dynamically remove a vote so that you have an odd number.

You can check dynamic weights in PowerShell. In this four-node cluster, I have allowed all four nodes a vote (NodeWeight), but the cluster has dynamically reduced the vote for the node ELVIS. That results in an odd number of total votes.



```
PS C:\> Get-ClusterNode | Format-Table name, dynamicweight, nodeweight, id, state -AutoSize

Name    DynamicWeight NodeWeight Id State
----    ------------- ---------- -- -----
CARL                1          1 2     Up
CHARLIE             1          1 3     Up
ELVIS               0          1 1     Up
ROGER               1          1 4     Up
```

Node Dynamic Weight is visible in the graphical interface of the Windows Failover Cluster Manager beginning in Windows Server 2012 R2, 2016, and 2019 and later version of Windows.

Here are the cluster quorum recommendations for Windows Server 2019 (x):

- If you have two nodes, a witness is required.
- If you have three or four nodes, a witness is strongly recommended.
- If you have Internet access, use a cloud witness
- If you are in an IT environment with other machines and file shares, use a file share witness

DYNAMIC COUNTING OF WITNESS VOTES; TIEBREAKER OPTION

The cluster can decide whether or not it should give the witness a vote based on the state of witness and the available number of nodes. It also provides more options, helping nodes in your primary datacenter stay online if you end up in a tied-vote situation between your primary and disaster recovery data centers.

RULES FOR DESIGNING QUORUM

Here are a few rules to keep you sane when setting up your cluster:

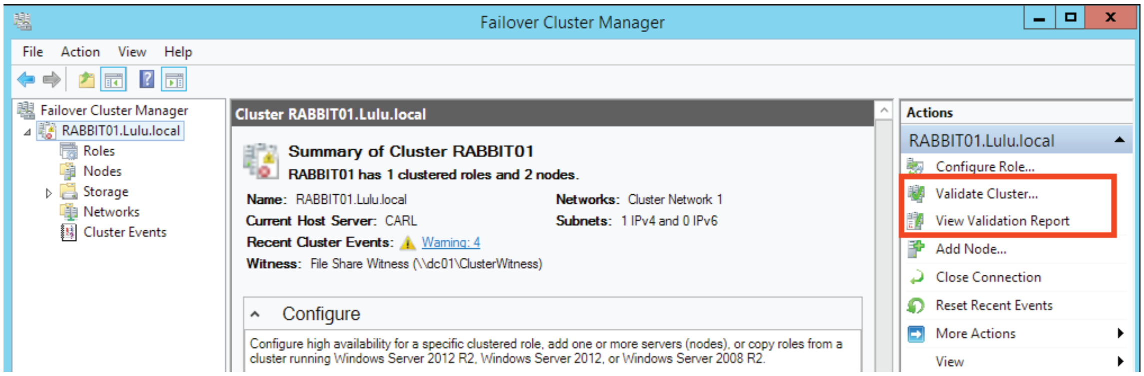
- Your goal is to have an odd number of votes: use a witness when necessary
- In Windows Server 2012 R2, 2016, and 2019, always use a witness and allow the cluster to manage the vote dynamically
- If your cluster stretches across sites and network communication is unreliable, consider removing votes from nodes in the secondary site
- Use a Disk Witness whenever possible (instead of a File Share Witness)
- Before going live, test out many failure scenarios to understand how your version of Windows handles dynamic voting.

MISTAKE #1: SKIPPING CLUSTER VALIDATION (OR IGNORING WARNINGS)

Cluster Validation is the most important part of the building and configuring a Windows Failover Cluster. Never skip this step. You should run a validation a few times:

- Validate the configuration before cluster install
- Run validation after the cluster is set up and you have completely installed and configured your SQL Server instances on the cluster
- Validate whenever you make changes to the cluster

Not all validation tests can be run online. Make sure validation results are perfect before going live to avoid having to schedule downtime.



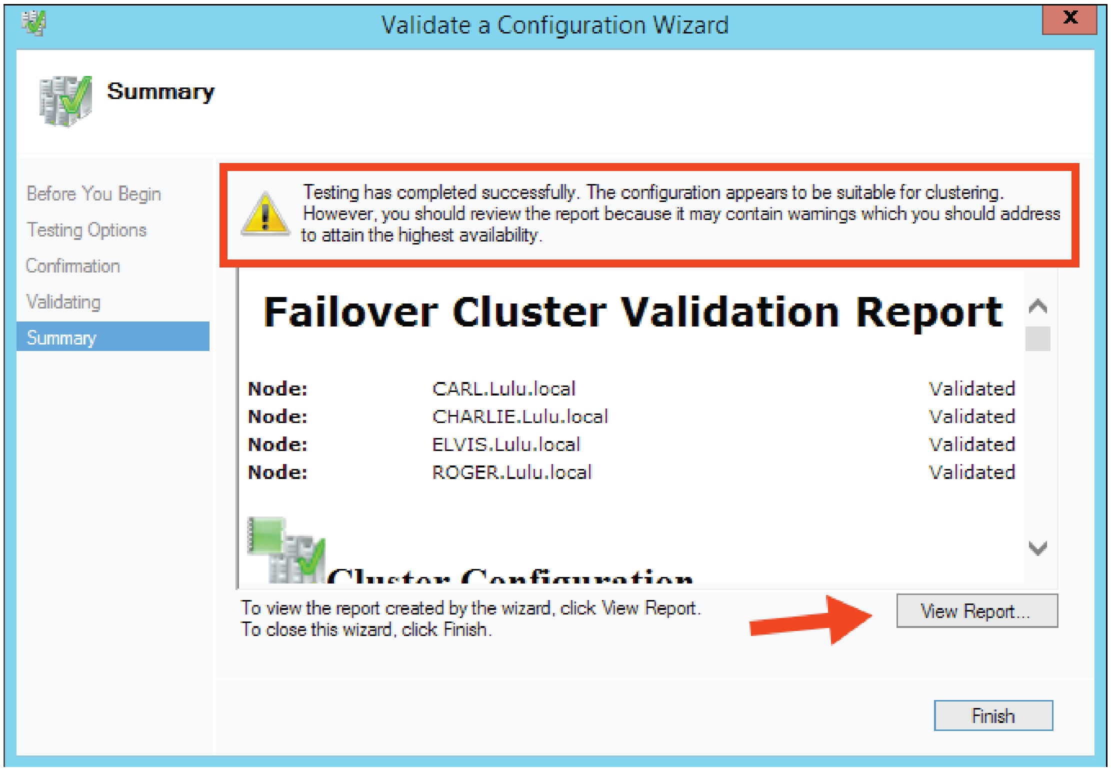
Summary of Cluster RABBIT01
RABBIT01 has 1 clustered roles and 2 nodes.

Name: RABBIT01.Lulu.local
Current Host Server: CARL
Recent Cluster Events: ⚠ Warning: 4
Witness: File Share Witness (\\dc01\ClusterWitness)
Networks: Cluster Network 1
Subnets: 1 IPv4 and 0 IPv6

Configure high availability for a specific clustered role, add one or more servers (nodes), or copy roles from a cluster running Windows Server 2012 R2, Windows Server 2012, or Windows Server 2008 R2.

TAKE EVERY WARNING OR ERROR SERIOUSLY

Validation is the primary way your cluster can warn you about problems in the configuration. Never assume that warnings can be skipped.



Testing has completed successfully. The configuration appears to be suitable for clustering. However, you should review the report because it may contain warnings which you should address to attain the highest availability.

Failover Cluster Validation Report

Node:	CARL.Lulu.local	Validated
Node:	CHARLIE.Lulu.local	Validated
Node:	ELVIS.Lulu.local	Validated
Node:	ROGER.Lulu.local	Validated

Cluster Configuration

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

Before you take the cluster into production use, make sure that you have run a full cluster validation and corrected every error and warning. If you do not correct a warning, document exactly why it is ignorable. Save a copy of your cluster validation reports off in a safe spot in case you need them for reference when trouble strikes.

WHAT IF YOU DID NOT BUILD THE CLUSTER?

Sometimes the Database Administrator who manages the cluster is not the same person who chose the hardware, installed the Windows Failover Cluster Feature, and added the nodes to the cluster. Perhaps the cluster was handed off to you, and you were told, “The cluster is already.”

Translation: What this means is that it is time for you to make sure that Cluster Validation is perfect.

YOU CAN AVOID THESE SETUP MISTAKES!

Let us face it: hardware will fail. Windows Failover Clustering is a great technology that can keep your SQL Server instances and databases online and working when the inevitable happens.

Armed with the knowledge in this whitepaper, you can set your next SQL Server Failover Cluster the right way.

FOOTNOTES

i. See “Licensing for High Availability” in the Microsoft SQL Server 2012 and 2014 Licensing Guides: 2012, 2014, 2016, 2017, and 2019.

ii. See “SQL Server Multi-Subnet Clustering”.

iii. See “Failover Clustering and Always On Availability Groups”.

iv. For example, for builds, see “What is the Smallest SQL Server You Should Build”.

v. See “Running Multiple Instances of SQL Server”. For many configurations, allowing only 1 to 2GB of physical memory to be free may not be sufficient for the operating system.

vi. See “Hardware Requirements for Failover Clusters”.

vii. Notable KBs for Windows 2008 R2 include: KB 2566205, KB 2468345, KB 2545635, KB 2553549 =http://technet.microsoft.com/en-us/library/cc771404.aspx

viii. See more information on Slipstreaming.

ix. See “KB 2494036”.

x. See “Understanding cluster and pool quorum”.

ABOUT THE AUTHOR

Kendra Little has been working with SQL Server for more than ten years. She has performance-tuned databases ranging from 1 gigabyte to 80 terabytes. Kendra is also a Microsoft Certified Master in SQL Server.