Tintri

TECHNICAL WHITE PAPER

SQL Server AlwaysOn Availability Groups (AG) on Tintri

Best Practices Guide

V1.1 – November 2015

Contents

Intended Audience

This Best Practices Guide for running SQL Server AlwaysOn Availability Groups on Tintri VMstore™ systems is intended to assist SQL DBAs, IT Administrators and Architects who are responsible for deploying and managing clustered Microsoft SQL Database servers within their virtual infrastructures powered by Tintri VMstore storage appliances. This guide focuses specifically on Clustering Microsoft SQL Server databases and is intended to be used as a supplement to the Microsoft SQL Server on Tintri - Best Practices Guide. Recommendations from the SQL Single Instance best practice guide should be followed prior to following SQL Server clustering recommendations found in this guide.

If you are not already versed in SQL clustering, we recommend you follow the content within this guide to create one or more SQL AlwaysOn Availability Groups in a test environment to gain familiarity with the technologies prior to jumping straight into a production deployment.

Executive Summary

Virtualization technology has matured significantly over the years and mission critical servers can safely be deployed as VMs. Deploying SQL Servers as Virtual Machines (VMs) provides higher availability and more flexibility than their bare metal physical server counterparts. While a standalone VM may be highly available with many 9's of uptime, the availability of the application that runs within the VM may not have the same high level of availability for reasons such as reboots required for OS & application patching, recovering from disasters and DR testing, to name a few. To obtain an even higher level of application availability, some administrators turn to clustering as a solution. This guide is intended to assist those who want to further minimize disruptions caused by application outage and provide higher levels of resiliency.

With SQL Server 2012, Microsoft introduced "AlwaysOn" Technologies including Server Failover Instances and Availability Groups, a new way to cluster databases that departs from the previous methods that relied on Microsoft Clustering Services (MSCS). MSCS has historically been difficult to configure within a virtual environment and often brought limitations due to its requirements on having Raw Disk Mappings (RDM). With the new SQL Server AlwaysOn Availability Groups (AG), MSCS is no longer required.

This guide will walk you through the process of setting up SQL Server Availability Group in a virtualized environment and configuring your databases to achieve higher availability than what could be achieved in the virtual infrastructure layer alone. For your convenience, many links to additional information have been provided in each section.

Not included in this guide: the storage configuration. This is NOT an oversight! There is no manual tweaking/tuning required on a Tintri VMstore to accommodate a SQL Workload. Whether SQL is configured as AG or stand-alone, the Tintri VMstore automatically adapts.

This guide was created using Microsoft SQL Server 2014 on Windows Server 2012 R2 VMs running on VMware vSphere 5.1 and 5.5 hosts, but the information within is also applicable to MS SQL Server 2012 on Windows 2008 R2 and/or Windows 2012 running on versions of vSphere that are compatible with Tintri VMstores.

Assumptions

This document assumes you are working with a fully configured, highly-available virtual infrastructure. The VMs used for SQL Server are configured as described in our Microsoft SQL Server on Tintri Best Practices Guide and you have reviewed the recommendations provided within that guide.

Microsoft SQL Server AlwaysOn Availability Groups Overview

The AlwaysOn Availability Groups (AG) feature is a high-availability and disaster-recovery solution that provides an enterprise-level alternative to database mirroring. Introduced in SQL Server 2012, Availability Groups maximize the availability of a set of user databases for an enterprise. An *availability group* supports a failover environment for a discrete set of user databases, known as *availability databases* that failover together.

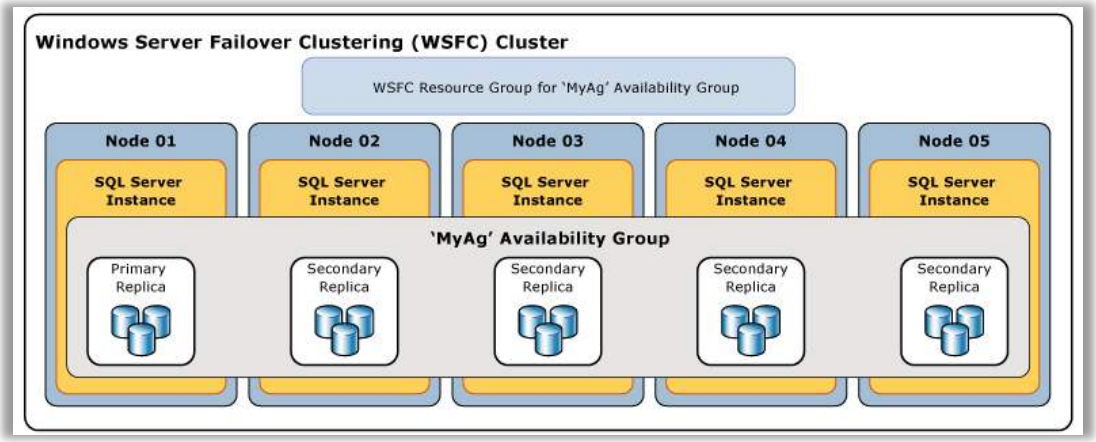


Figure 1 - Graphical overview of SQL Server AG. Image source: http://msdn.microsoft.com/en-us/library/ff877884.aspx

An availability group supports a set of read-write primary databases and up to eight sets of corresponding secondary databases in SQL Server 2014 (only up to four secondary DBs in SQL 2012). Optionally, secondary databases can be made available for read-only access and/or some backup operations. Microsoft's article Windows Server Failover Clustering (WSFC) with SQL Server describes two methods of applying AlwaysOn Technologies to SQL Server:

- **Database-level High Availability with AlwaysOn Availability Groups (AG)**
  - **Availability Groups are SUPPORTED ON TINTRI**
  - Does NOT require shared disk.
  - Some reporting tasks can be offloaded to the replica instances.
  - Backup operations can be offloaded to the secondary replica instances. This will minimize load on the primary database.
  - Think of this method as **"Data Availability".**
- **Failover Cluster Instances (FCI) combined with AlwaysOn Availability Groups (AG)**
  - Failover Cluster Instances (FCI) are **NOT YET** supported on Tintri.
  - Requires shared disk that supports the SMB 3.0 Remote Shared Virtual Disk protocol.
  - SQL Server application runs on one node of a 2 node cluster. The other node is passive awaiting failover from the primary node, should it fail.

This guide has been written to address Database-level AlwaysOn Availability Groups (AG).

***DO**: Use Database-level High Availability with AlwaysOn Availability Groups (AG)*

***DON'T**: Use Failover Cluster Instances (FCI) with the Tintri VMstore providing the shared storage.*

***DO:** Leverage AG read-only secondaries as a way to decrease overall vCPU footprint per VM*

Consolidated List of Practices

The table below includes the recommended practices in this document. Click the text on any of the recommendations to jump to the section that corresponds to each recommendation for additional information.

DO: Use Database-level High Availability with AlwaysOn Availability Groups (AG)

DON'T: Use Failover Cluster Instances (FCI) with the Tintri VMstore providing the shared storage.

DO: Leverage AG read-only secondaries as a way to decrease overall vCPU footprint per VM

DO: Create an AD security group to simplify management of permissions required for the service accounts of the SQL Server Service on each node prior to setting up SQL Server AG databases.

DO: Reboot SQL Server VMs and/or restart the SQL Server service prior to proceeding in the Configuration section if group memberships to the AD security group were recently changed.

DO: Use a file share as a Quorum and ensure that all nodes have sufficient permissions to read and write to the share.

DO: Consider deploying the file server as a VM. Ensure that the file share is highly available and that the file server VM hosting the file share is highly available.

DO: Size VMs participating in a SQL Availability Group identically

DO: For future changes to a VM's configured size, remember to also resize the other VMs with nodes participating in the same SQL Server AG.

DO: Keep SQL Server AG VMs on separate Tintri VMstores for Highest Availability.

DO: Review the warnings and recommendations within the results of the Cluster Validation Test. Determine whether they are valid concerns, or false positives (i.e. Disk missing – not applicable since we are not using a shared disk configuration)

DO: When prompted with the Confirmation screen during the Cluster Creation Wizard, be sure to UNCHECK the "Add all eligible Storage to the cluster" checkbox. Failing to do this may render existing volumes inaccessible.

DO: Configure the quorum options for "Select the quorum witness" and provide the path to the file share to be used as the quorum witness.

DO: Create a rule to prevent SQL Server VMs that are members of the same Availability Group (AG) from running on the same host.

Prerequisites

Licensing - OS and Application Editions

SQL Licensing - SQL Server Enterprise Edition is required to leverage AG with SQL Server. The article Features Supported by the Editions of SQL Server 2014 provides details on various licensing options of SQL Server.

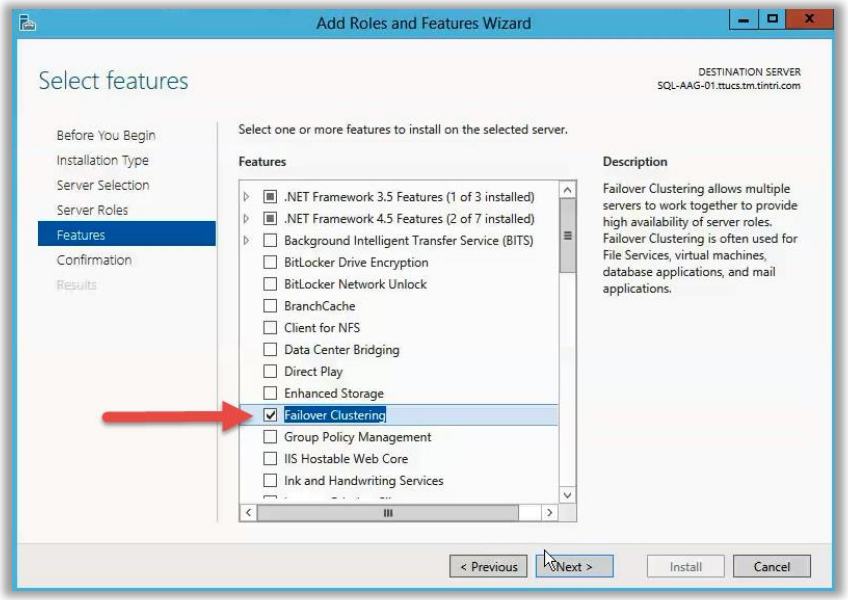
Windows Licensing - SQL Server AG requires two main Roles/Features that are not common across all licensed editions of Windows Server. These are Windows Failover Clustering (WSFC) feature and the Application Server role. Both of these features combined are only available with licenses for the following versions:

- Windows Server 2008 R2: Datacenter and Enterprise Editions only.
- Windows Server 2012 / 2012 R2: Datacenter and Standard Editions only.

In a virtual infrastructure with high VM to host consolidation ratios, Datacenter licensing is more cost-effective and applied per-CPU socket of each host.

Refer to Windows Server 2012 R2 Products and Editions Comparison for more information.

Windows Failover Clustering (WSFC)



SQL Server AG leverages technology made available with the Windows Failover Clustering (WSFC) feature, found only in versions of Windows Server listed above. Before we are able to configure WSFC, we must first add the feature to each of the VMs we would like to include in our SQL Database Cluster(s). To do so, use the Add Roles and Features Wizard to add the “Failover Clustering” feature to each of your VMs, as shown here:

Figure 2 - Add the Failover Clustering feature to each VM

Active Directory (AD) Permissions & Security

Configuring a failover cluster within an AD environment (the method described in this guide) requires the SQL Server VMs to be members of the same Windows Domain. Domain Admin-level access permissions are required to get it setup and configured. If you do not have Domain Admin access rights, refer to this article to Pre-stage Cluster Computer Objects in Active Directory Domain Services , which also contains useful information on the exact permissions required.

Introduced in Windows 2012 R2 is the ability to Deploy an Active Directory-Detached Cluster, which removes the AD requirement. The method in this article is untested by Tintri and out of scope of this guide.

SQL Server Accounts, Permissions and Groups

Throughout this guide, references are made to ensuring that the SQL Server nodes have sufficient permissions to various AD objects, SQL objects, and other resources. Most often, this refers to the Service Account used on the SQL Server Service, as shown here:

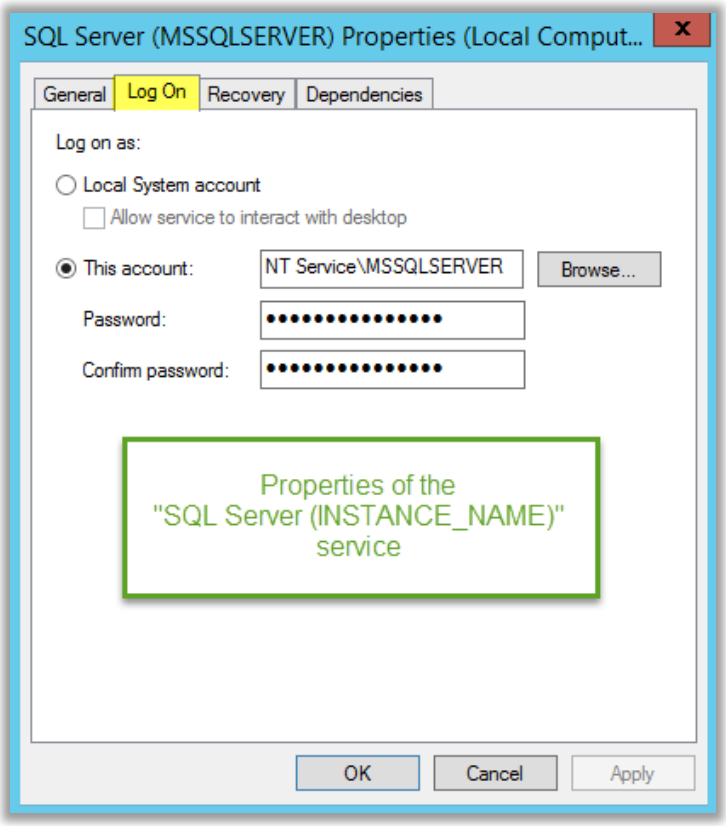


Figure 3 - Use the Log On tab on the properties of the SQL Server service to view or change the service account

Configuration and testing carried out for the creation of this guide used the default service account for SQL. To other systems on the network, this account appears as the computer domain account of the SQL Server Virtual Machine: *DOMAIN\MACHINENAME$*. Because two or more nodes are used, all ACLs for resources being permissioned need to include all the nodes.

To simplify management of permissions for the computer accounts, an AD security group named “DOMAIN\SQL-AAG-Computers” is created. Membership of this group contained both SQL Server computer accounts used in the cluster, as well as Computer Object for the cluster (to be created later):

***DO**: Create an AD security group to simplify management of permissions required for the service accounts of the SQL Server Service on each node prior to setting up SQL Server AG databases.*

In the future, if a new node is added to the cluster, simply add the new computer account or service account to this group. In your organization’s environment, one or more unique AD Service Accounts (User objects) may be used for the “Logon As” properties of the SQL Server service instead of the default machine account.

Throughout this guide, use this newly created security group for applying permissions to the various resources (file share, SQL permissions, computer objects, etc… which are covered later).

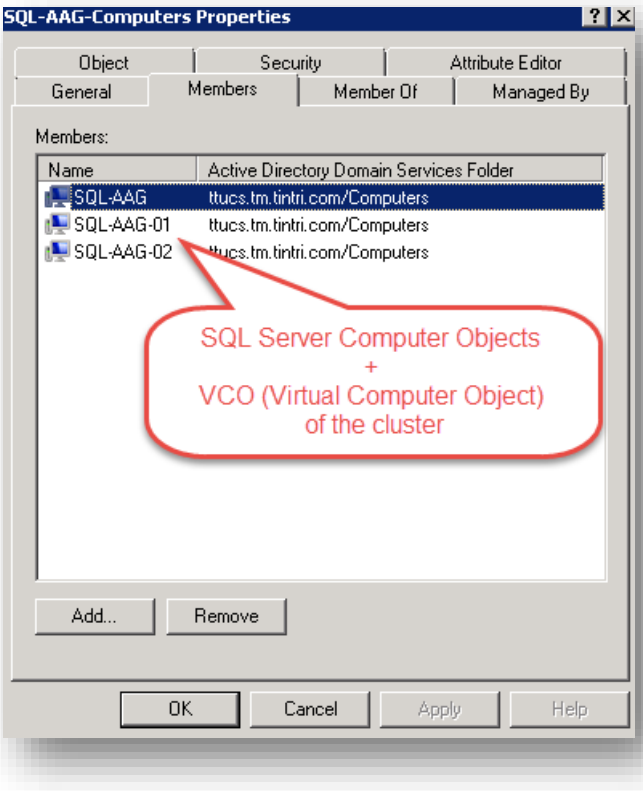


Figure 4 - AD Group created to simplify managing permissions

NOTE: Computer accounts won’t register as members of a new security group until AFTER a reboot, which is required to build a new token containing this group for the computer account. In the case of AD service accounts, services need to be restarted for the accounts to register as members of the new security group.

***DO**: Reboot SQL Server VMs and/or restart the SQL Server service prior to proceeding in the Configuration section if group memberships to the AD security group were recently changed.*

For each computer account of nodes that will join the cluster (DOMAIN\SQL-AAG-01$ and DOMAIN\SQL-AAG-02$), the security group (DOMAIN\SQL-AAG-Computers) was added to the security tab and given full-control:

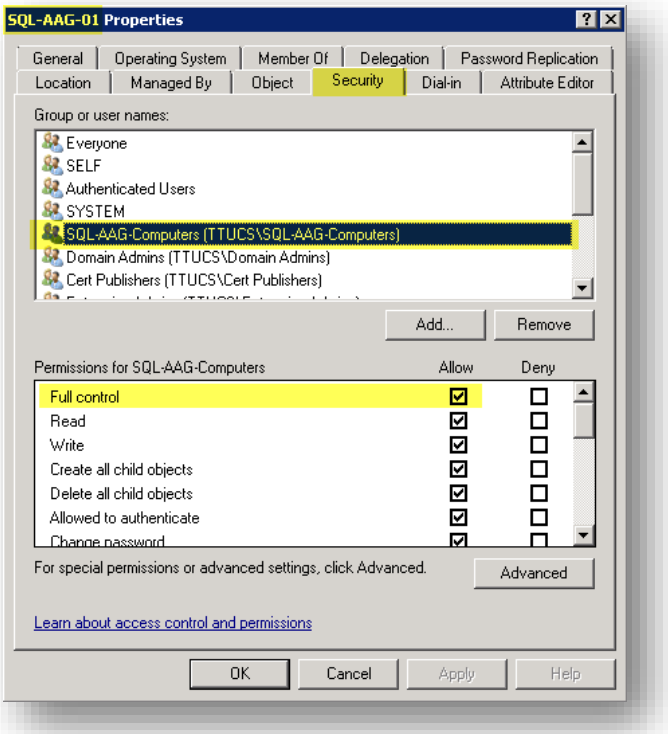
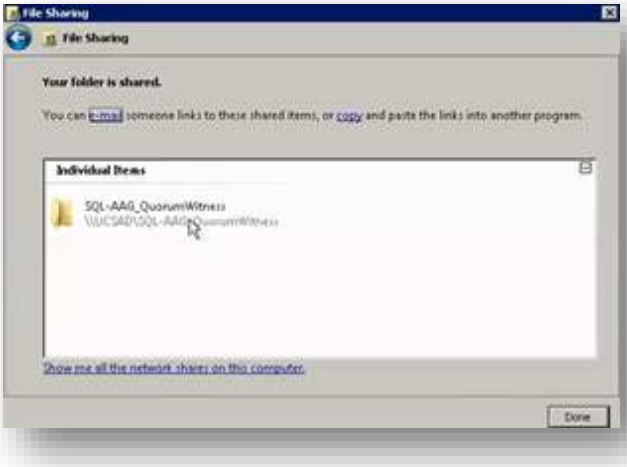


Figure 5 - Security group added to Computer Account (or Service Accounts)

Create a File Share for Quorum

To prevent a cluster of two nodes (or any ***even*** number of nodes) from suffering split-brain, a condition where both nodes believe they are the master, a quorum is used. The quorum can be thought of as a 3rd party tie-breaker. Traditionally, this has been a shared disk accessible to both nodes, but with recent improvements in WSFC, a file share can be used instead.



Create a File Share to use as a Quorum for the failover cluster. The file share must be network-accessible to all nodes of your cluster, and permissions must be configured to allow all nodes to read and write from the file share. **Use the security group you created**

***DO**: Use a file share as a Quorum and ensure that all nodes have sufficient permissions to read and write to the share.*

Figure 6 - A file share has been created on a 3rd Party server to be used as a quorum

Use an existing file server on which to create the file share or consider creating a new dedicated VM for the share. Availability of the quorum (file share) can adversely affect the health of your failover cluster, so be sure that your file share is also highly available.

***DO**: Consider deploying the file server as a VM. Ensure that the file share is highly available and that the file server VM hosting the file share is highly available.*

Virtual Machine Sizing

Before proceeding further within this guide, refer to our Microsoft SQL Server on Tintri Best Practices Guide and apply the recommendations to your SQL VM(s).

When sizing SQL VMs that will participate as nodes in the same Availability Group, we recommend you configure and size them all identically, including:

- # of vCPUs
- Memory
- # of vNICs (2 or more recommended, with one of the vNICs being dedicated to WFSC traffic)
- # of vDisks, with matching capacities, SCSI controller properties, and SCSI IDs
- any other property of the VM specifically tuned for SQL

There may be exceptions to this recommendation where you may want to size VMs from the same Availability Group differently, specifically in cases where one or more of the secondary instances will be used as a read-only copy of the database(s). Use cases for this include read-only copies for reporting, exporting, and backup services. We recommend that you have at least two VMs sized the same that are capable of handling the full production load in the event of a VM failure, but additional read-only instances (up to eight are allowed with SQL 2014) may be sized smaller, depending on their use case.

***DO**: Size VMs participating in a SQL Availability Group identically*

By offloading some of the load associated with reporting and backup processes, it is possible to size your primary SQL Servers smaller (fewer vCPUs) than if a single VM had to handle the load of all user and application access, reporting and backups. Smaller VMs schedule better in a virtual infrastructure and are less likely to create resource contention, specifically where vCPU counts within a VM are high.

Here are some additional tips to consider with respect to VM sizing:

- If you extend the capacity of a vDisk on one VM, be sure to extend the same corresponding drive on all other VMs with the same AlwaysOn Databases
- When increasing RAM on a VM, remember to not only adjust the other VM(s), but also to extend the size of the vDisk allocated to the Page file on the primary VM, as well as the secondary instances. The vDisk assigned to hold the page file should be at least the same size as the amount of RAM allocated to the VM.
- Use multiple smaller VMs with fewer vCPUs per VM and leverage read-only secondary SQL Server AG copies to handle specific reporting demands. Reducing the overall CPU demand from the production SQL servers may allow you to decrease their vCPU count to improving performance and scalability holistically with the virtual environment.

***DO**: For future changes to a VM's configured size, remember to also resize the other VMs with nodes participating in the same SQL Server AG.*

Different Sites and VMstores (Optional)

Failover Clustering may be used to enable High Availability within the same site to maintain uptime during application and OS patching. Or it may be used as a Disaster Recovery (DR) solution to provide failover to another site. And it may also be used serve both use cases, with two instances in the primary site and a third instance in the DR site. There are many possibilities to consider while architecting your solution, but keep in mind that the primary goal of SQL Server AG configurations is to provide maximum uptime and mitigate the risk of failure of common components. Spreading instances across Tintri VMstores is an effective way to mitigate the risk of a VMstore becoming unavailable (power loss, physical damage, human error, etc.).

***DO**: Keep SQL Server AG VMs on separate Tintri VMstores for Highest Availability.*

Additional Prerequisites and Information

For a more details, refer to Microsoft's article on Prerequisites, Restrictions, and Recommendations for AlwaysOn Availability Groups. When reviewing the link, keep in mind that it was written for both Database Availability and Failover Cluster Instances (FCI) and information on the latter (FCI) is not applicable as it is not support on Tintri VMstores, nor is it covered within this guide.

Configuration

In this section, we'll provide information on configuring SQL Server AG. It is assumed that you've met the prerequisites from the previous section and are starting with a Stand-alone instance of SQL, ready to extend it to an additional node for higher availability. Similar steps can be followed during setup of new SQL instances.

Configuring Windows Failover Clustering (WSFC)

Validate Configuration

With the WSFC feature added to each of the VMs we want to cluster, we are ready to perform a validation prior to creating a cluster. Open the Failover Cluster Manager, which can be accessed from the Tools menu of the main Server Manager dashboard in Windows 2012:

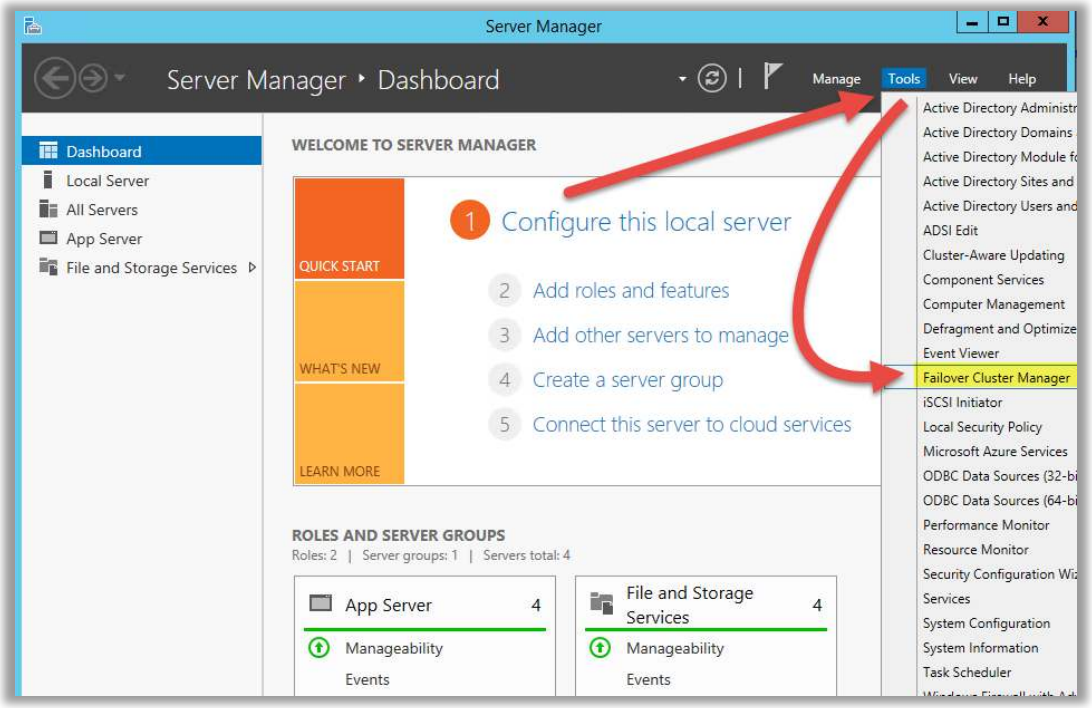


Figure 7 - Accessing the Failover Cluster Manager

Detailed Step-by-Step instructions for the validation process are found in Appendix A, however we'll cover some the highlights from the process in this section. From within the Failover Cluster Manager, start the configuration validation process:

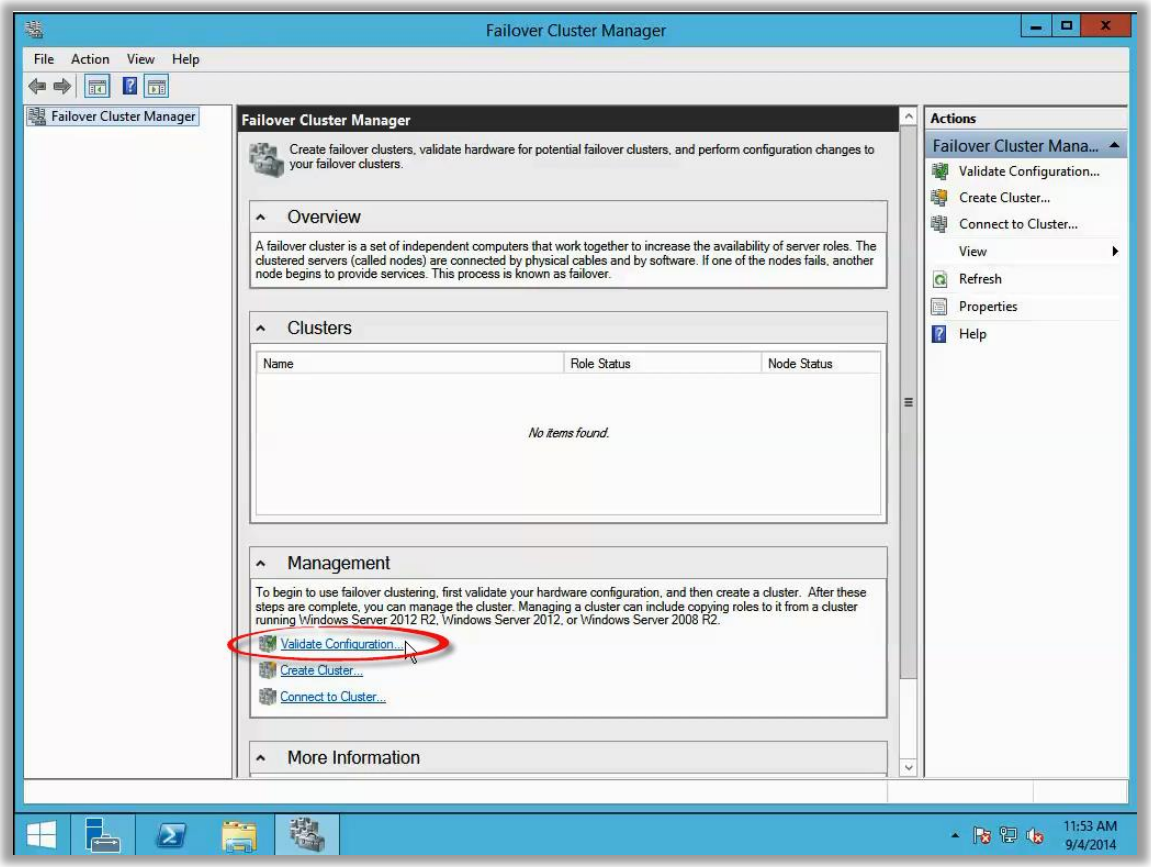


Figure 8 - Click on Validate Configuration

Add Servers to your cluster and run all tests.

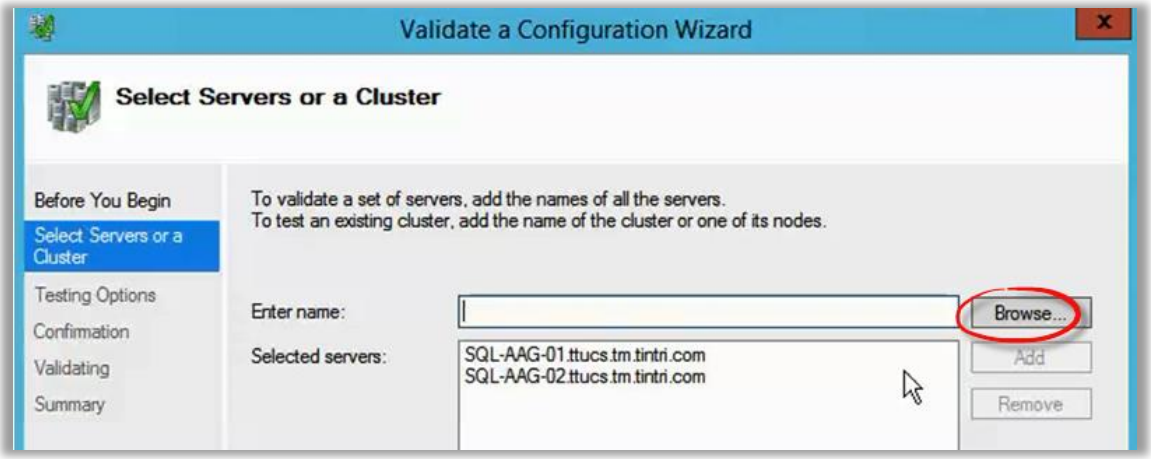


Figure 9 - Select Servers you want to add to a cluster using the Browse button, then click Next

After validation tests have run, a summary of test results will be show. Click on “View Report…” for a detailed summary of test results and recommendations:

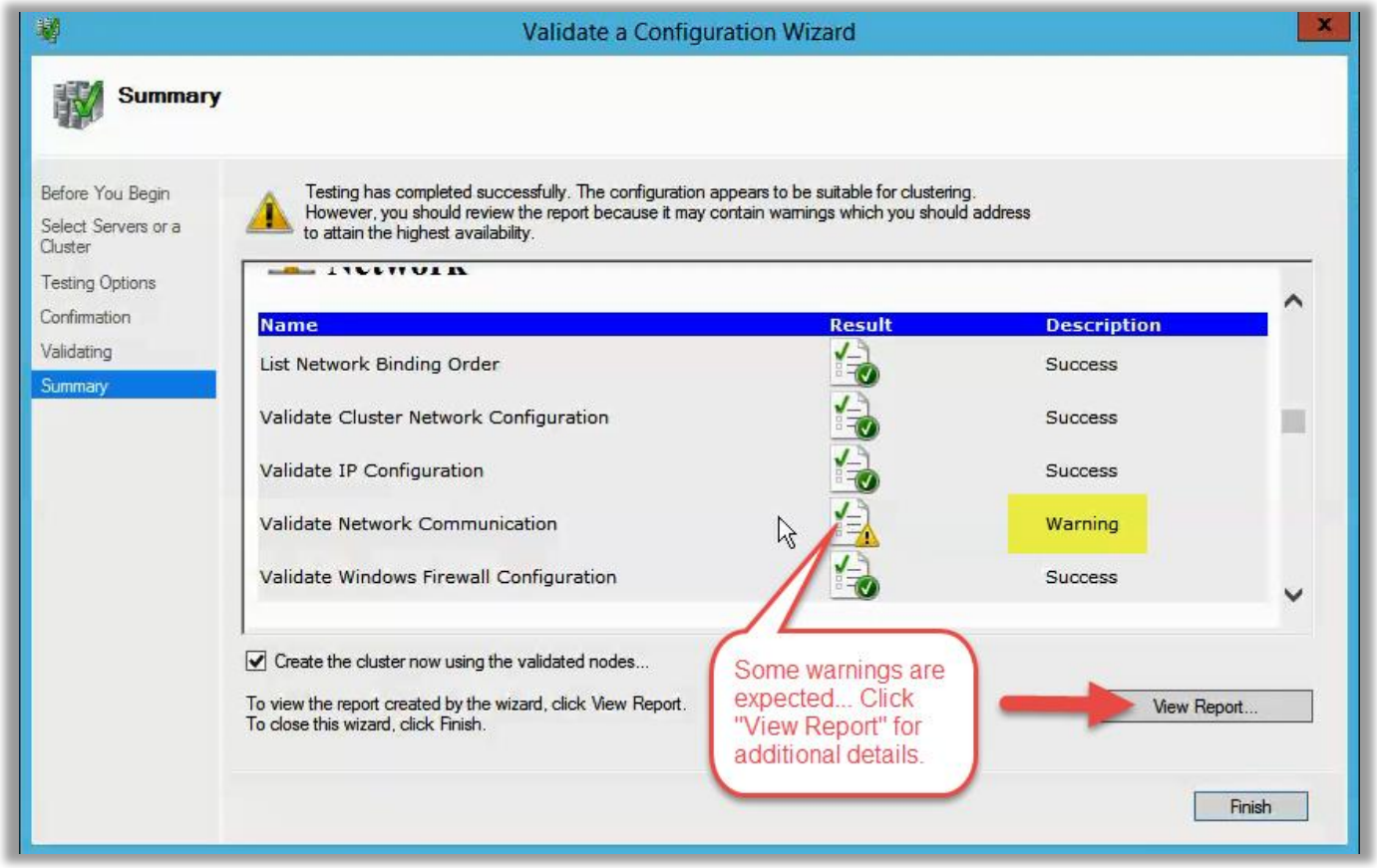


Figure 10 - Summary of Validation results

Some warnings are expected, such as the warning regarding networking redundancy as well as warning regarding storage.

***DO**: Review the warnings and recommendations within the results of the Cluster Validation Test. Determine whether they are valid concerns, or false positives (i.e. Disk missing – not applicable since we are not using a shared disk configuration)*

The warning about the number of NICs is likely not be accurate in the case of a virtual machine because the redundancy is built into the host design, but the guest OS (Windows) wouldn’t be able to detect that.

However, even though this may be the case, a 2nd NIC is recommended to isolate WSFC traffic.

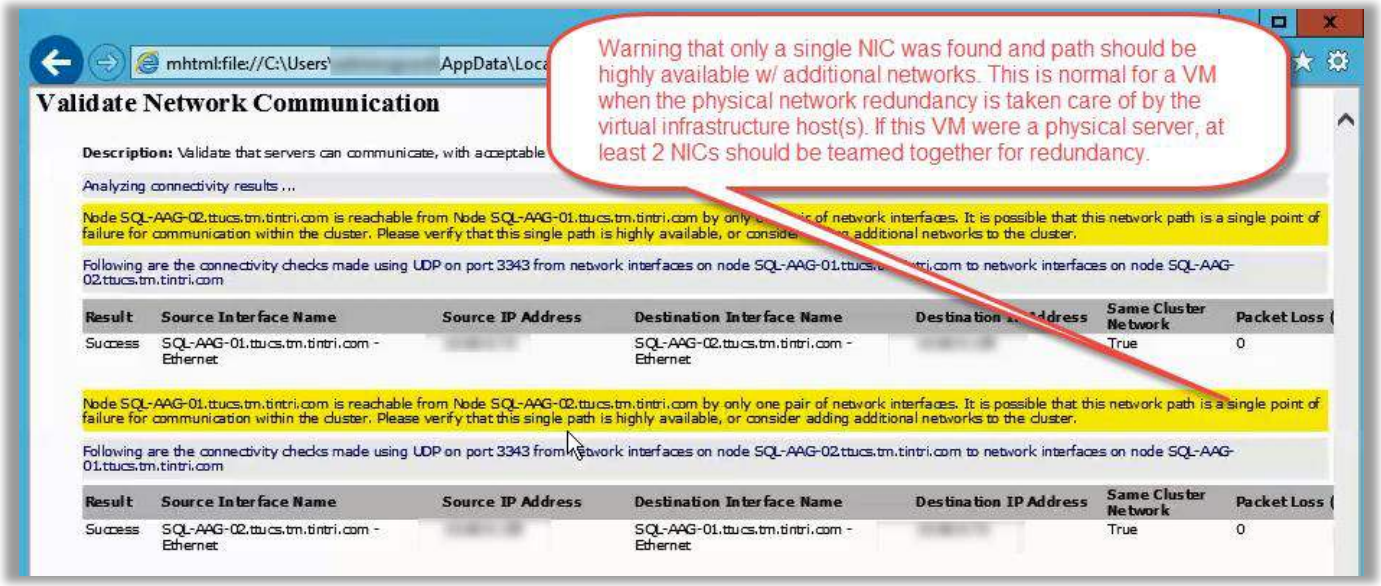


Figure 11 - Warning re: Networking Redundancy

There will be many storage warnings as well. Review each, but in most cases, these warnings shouldn't be applicable because we will not be configuring clustering that requires a shared disk.

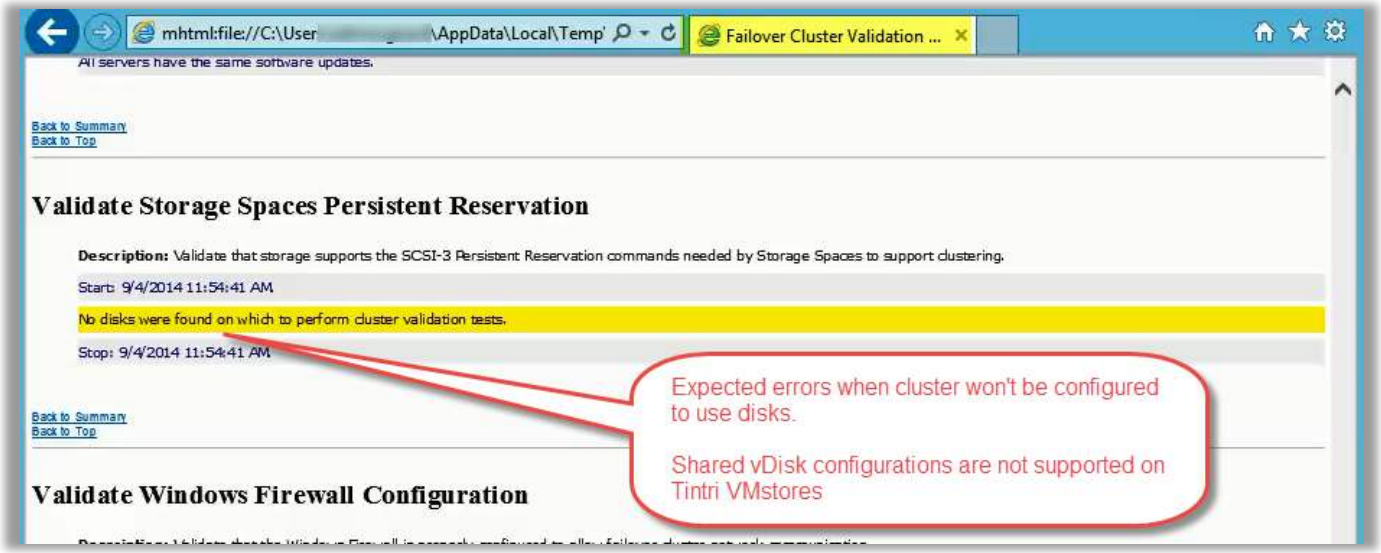


Figure 12 - Warnings re: disks are a false positive, and not applicable because we won't be using shared disks

Create a Windows Failover Cluster

Once the server configuration has been validated and warnings have been reviewed (and rectified or dismissed as false positives), we're ready to proceed with creating a cluster. Leave the "Create the cluster now using the validated nodes" option selected at the end of the validation wizard, then click Finish to launch the Create Cluster Wizard. Detailed Step-by-Step instructions are available in Appendix A, continuing on from the prior validation steps.

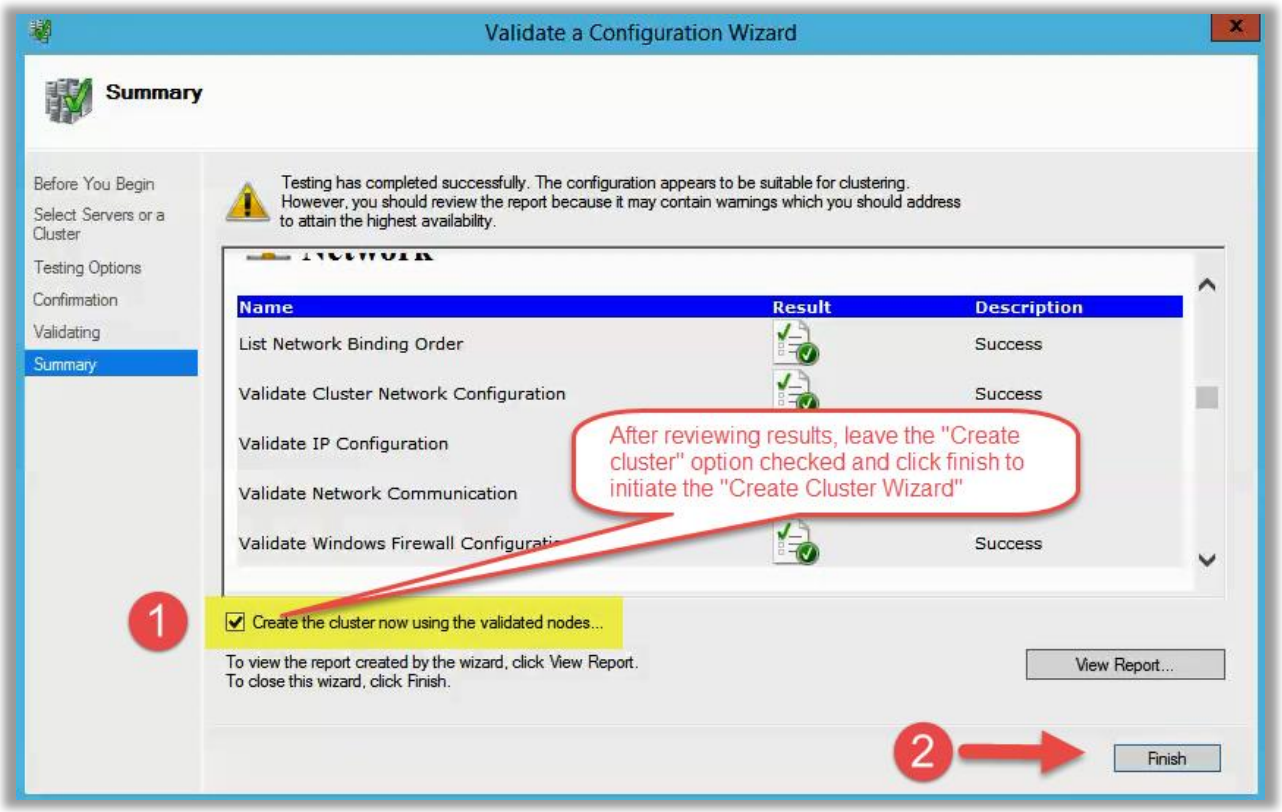


Figure 13 - Click Finish to launch the Create Cluster Wizard

Provide a name for the Failover Cluster:

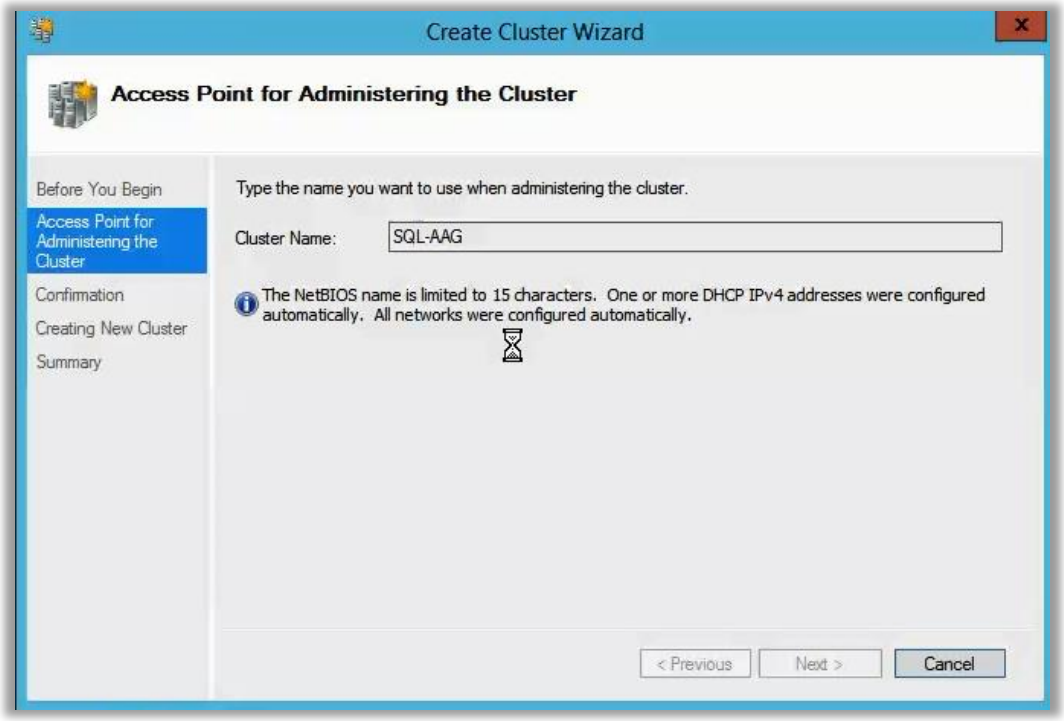


Figure 14 - Provide a name for your cluster. This will be the name of the virtual IP of the active cluster node

The name supplied here will create a Computer Object in the AD, in the default Computers OU, as shown here:

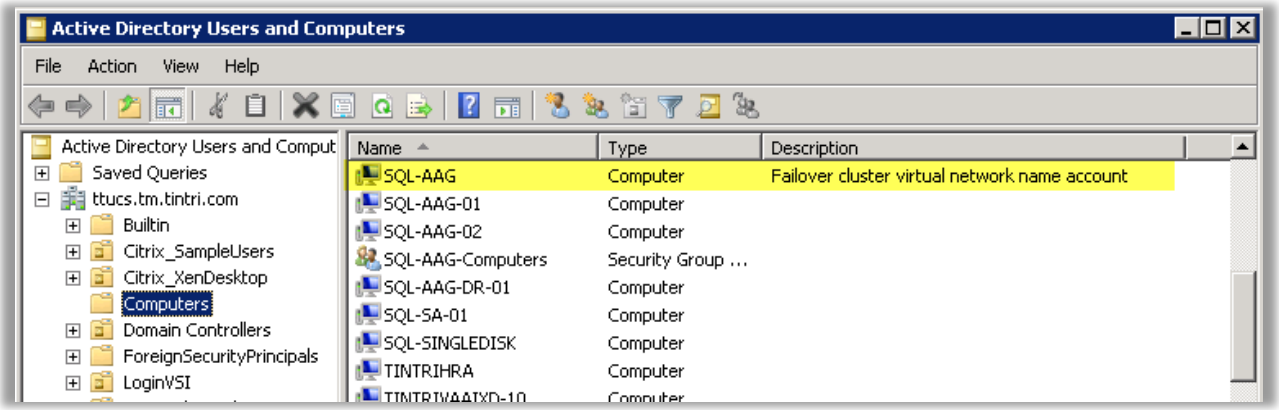


Figure 15 - A new Computer Object is created in the AD with the name provided in the Wizard

When users and applications access the Highly Available SQL AlwaysOn databases, this is the object that is entered as the SQL Server name, not the names of each underlying SQL Server instance (except in cases where a readable secondary wants to be explicitly accessed). In this case, users connect to **SQL-AAG**, which will send the request to the underlying server acting as master.

After providing a cluster name, a confirmation screen is shown. It is important to **uncheck** the “Add storage” checkbox, which defaults to checked, prior to clicking next to proceed, as shown here:

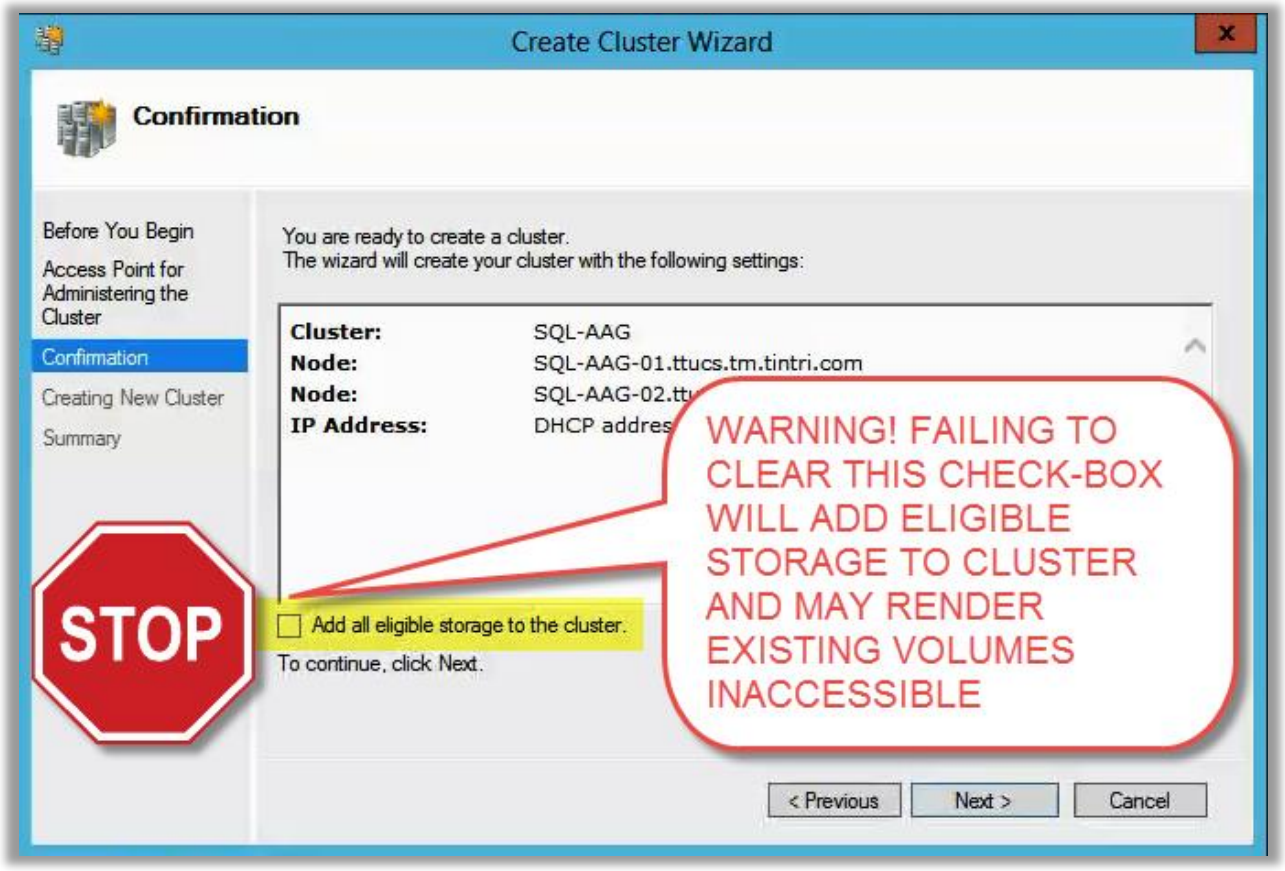


Figure 16 - On the confirmation, UNCHECK the "Add all eligible storage to the cluster" checkbox

---

***DO**: When prompted with the Confirmation screen during the Cluster Creation Wizard, be sure to UNCHECK the "Add all eligible Storage to the cluster" checkbox. Failing to do this may render existing volumes inaccessible.*

After the cluster has been created, click **View Report** in the final summary screen and review the details.

Configure Cluster Quorum Settings

Once the cluster has been successfully created, we need to configure a File Share Witness to be the cluster quorum. Detailed Step-by-Step instructions for this process can be found in Appendix B.

Instead of configuring a shared disk as our quorum, which is most commonly configured in a virtual environment using a Raw Disk Mapping (RDM) that can limit some VM functionality, we're going to configure a file share instead. In the prerequisites section, we created a file share and configured the permissions on it. To begin, launch the "Configure Cluster Quorum Settings" wizard from within Failover Cluster Manager, show here:

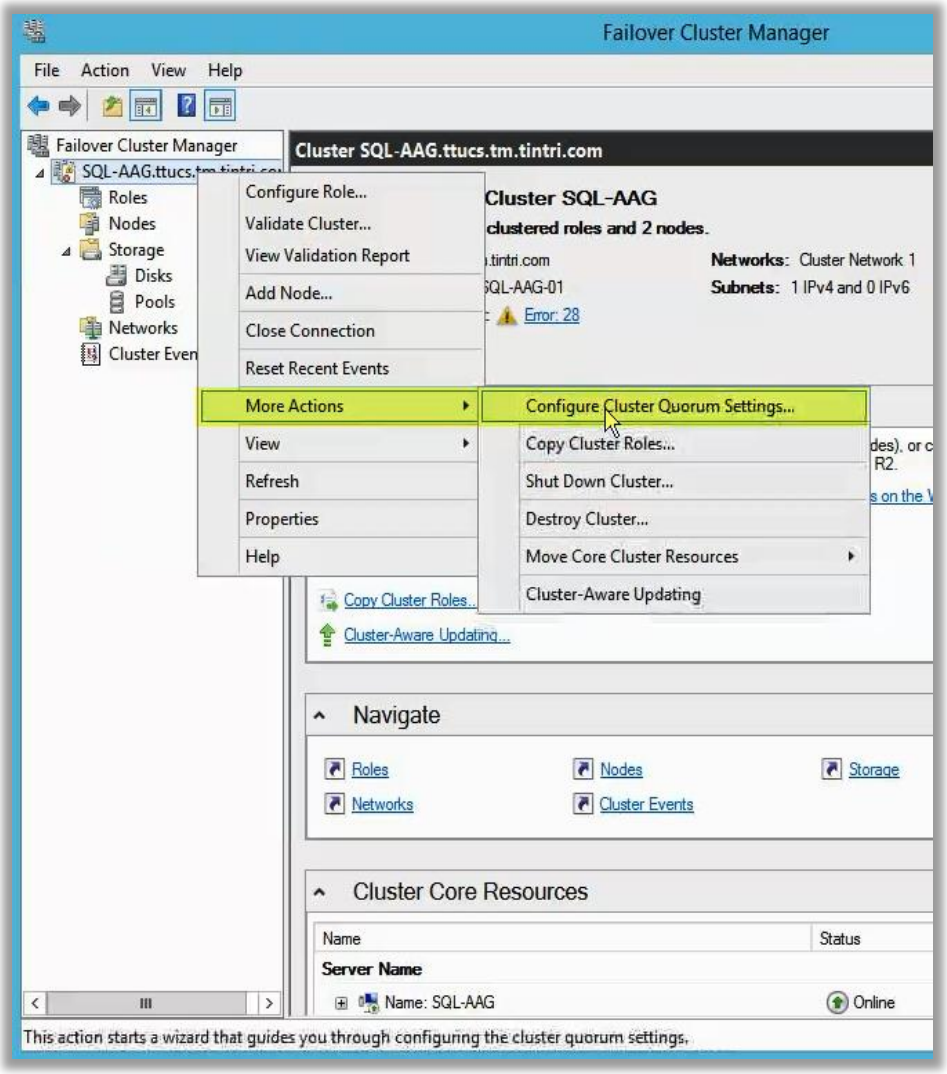


Figure 17 - Launch the Cluster Quorum Configuration Wizard

Choose the second option "**Select the Quorum Witness**" (shown below). Press **Next** to continue, and then provide the path to the file share, \\FULLY.QUALIFIED.HOSTNAME\Sharename, when prompted.

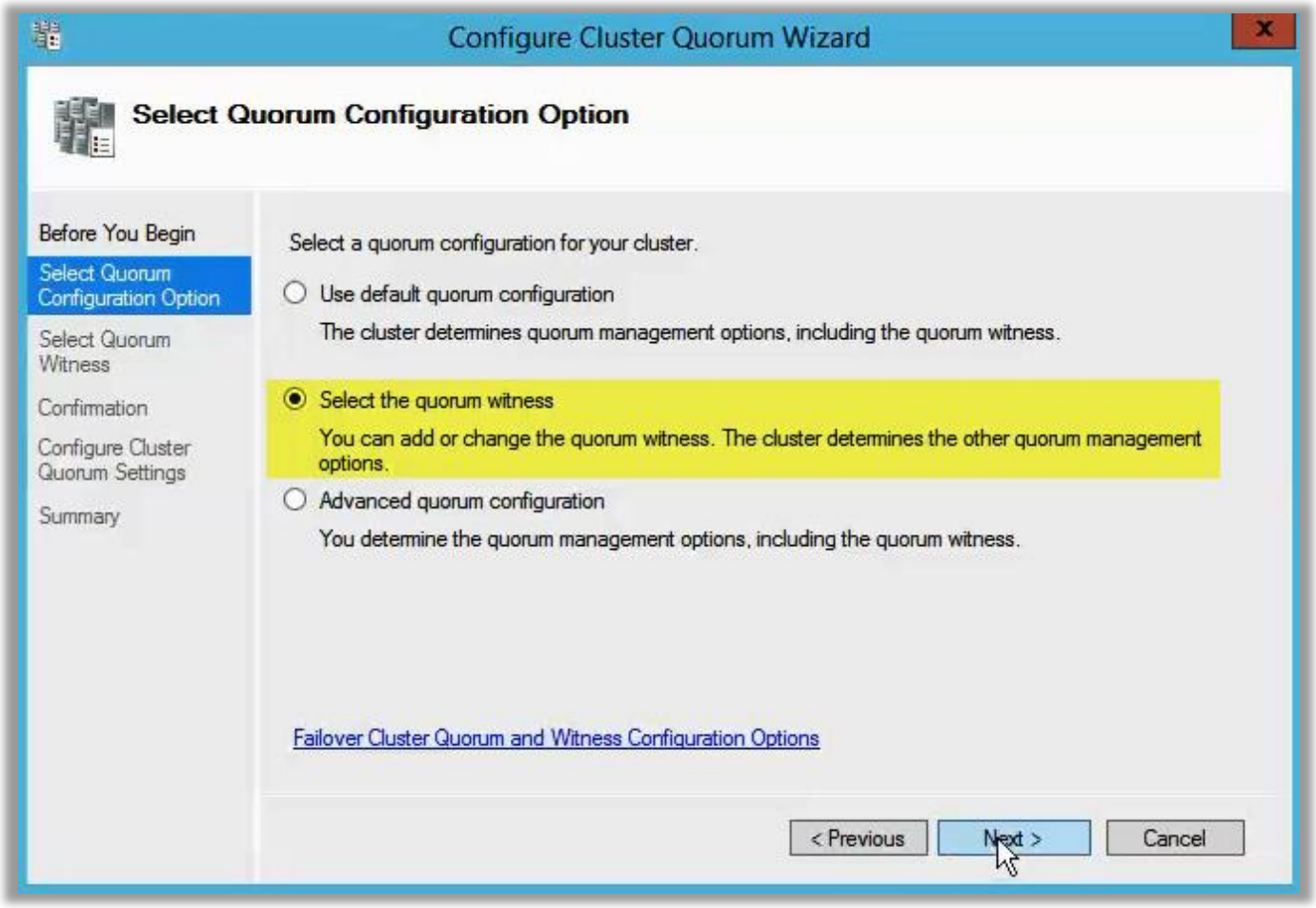


Figure 18 - Cluster Quorum Configuration Wizard - Select Quorum Witness

---

***DO**: Configure the quorum options for "Select the quorum witness" and provide the path to the file share to be used as the quorum witness.*

Tip: When using an asymmetric storage configuration for Availability Groups, you should generally use the **Node Majority** quorum mode when you have an odd number of voting nodes, or the **Node and File Share Majority** quorum mode when you have an even number of voting nodes. For the majority of this guide, we've been referencing 2 nodes and testing was performed with **Node and File Share Majority quorum mode**. WSFC Quorum Modes and Voting Configuration

SQL Permissions

In order to configure SQL Server clustering, each node will require some level of access to other nodes. The security group (DOMAIN\SQL-AAG-Computers) simplifies this task for us. On each SQL Server instance, use the SQL Server Management Studio to create an ID for domain-based security group that contains SQL Server service accounts.

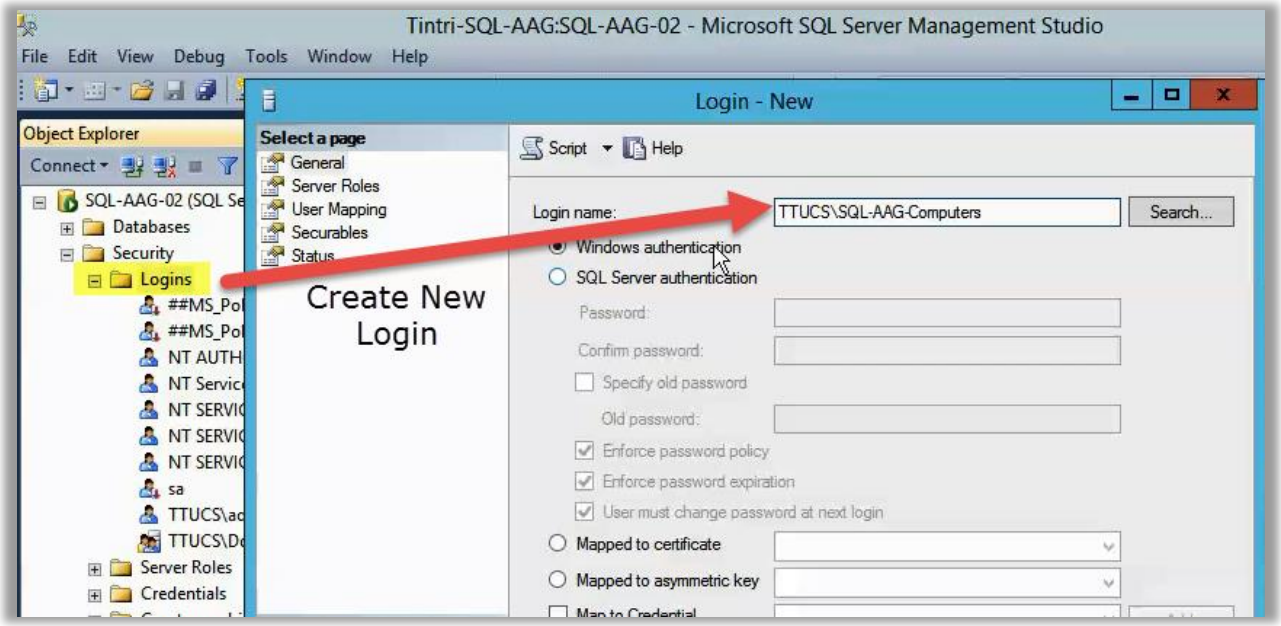


Figure 19 - Create a SQL Login for the Security Group contain Service Accounts

The new login for the security group only needs minimal CONNECT privileges (Public role).

Once the new login is created, assign CONNECT permissions to the HA_DR Endpoint on EACH SQL Instance, using the following command:

```
use [master]
GO
GRANT CONNECT ON ENDPOINT::[Hadr_endpoint] TO [DOMAIN\SQL-AAG-COMPUTERS]
GO
```

By proactively assigning this permission to the endpoint object on each server, you may be saving yourself hours of troubleshooting by avoiding the problems described in this article: Failed to Join the Instance to the Availability Group while configuring AlwaysOn and avoid getting this error in the AG Creation Wizard (next section):

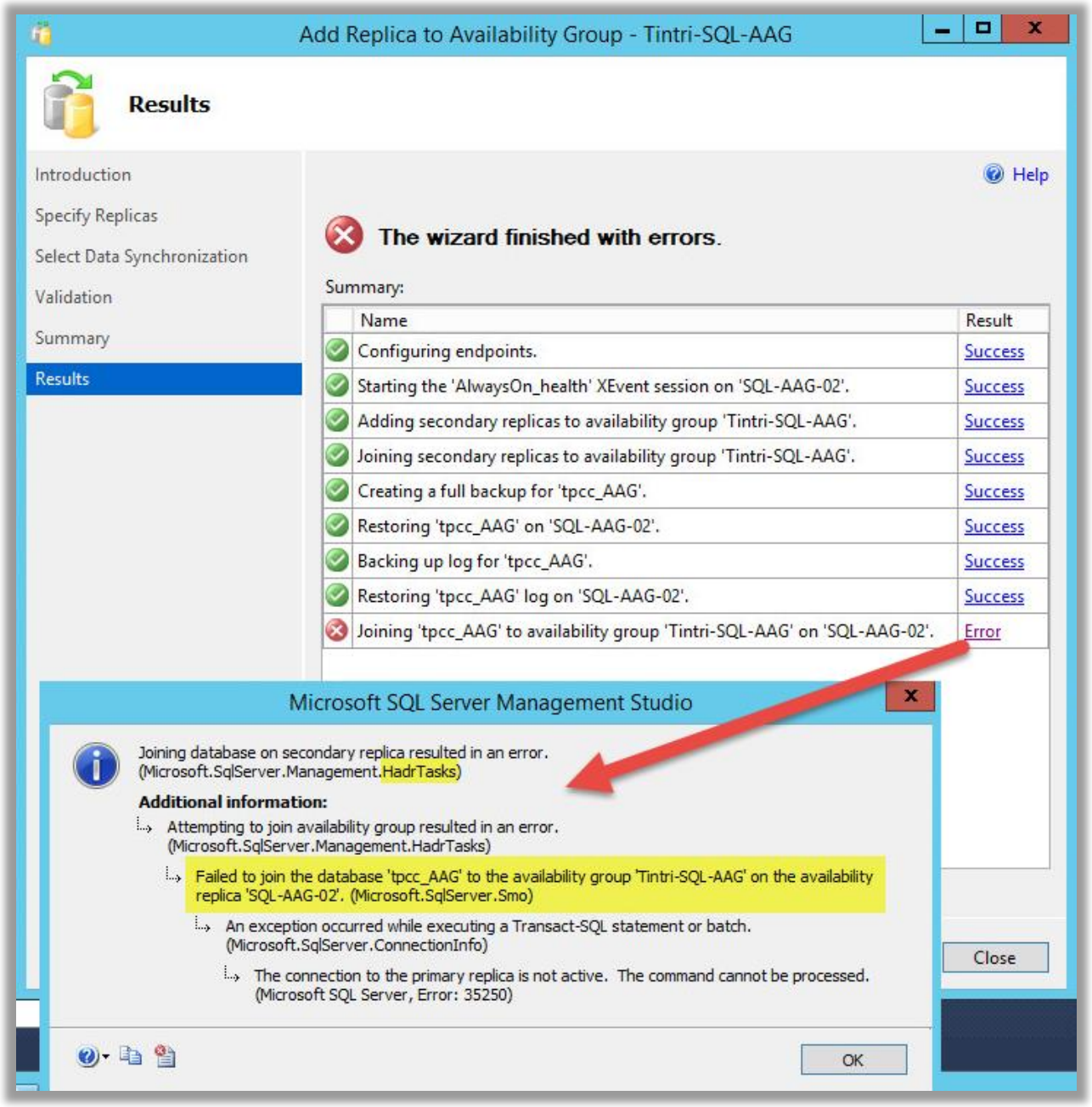


Figure 20 - An error due to not having the required permissions in place

More information on this setting can be found here: [Setup Login Accounts for Database Mirroring or AlwaysOn Availability Groups]

Enable SQL Server Availability Groups

At this stage, the underlying Failover Clustering is configured on your VMs and you are ready to enable SQL Availability Groups within SQL Server. Here are the steps:

1. Open SQL **Server Configuration Manager** and navigate to **SQL Server Services**.
2. **Right-click** on the **"SQL Server (INSTANANCE_NAME)"** service and choose **Properties**.

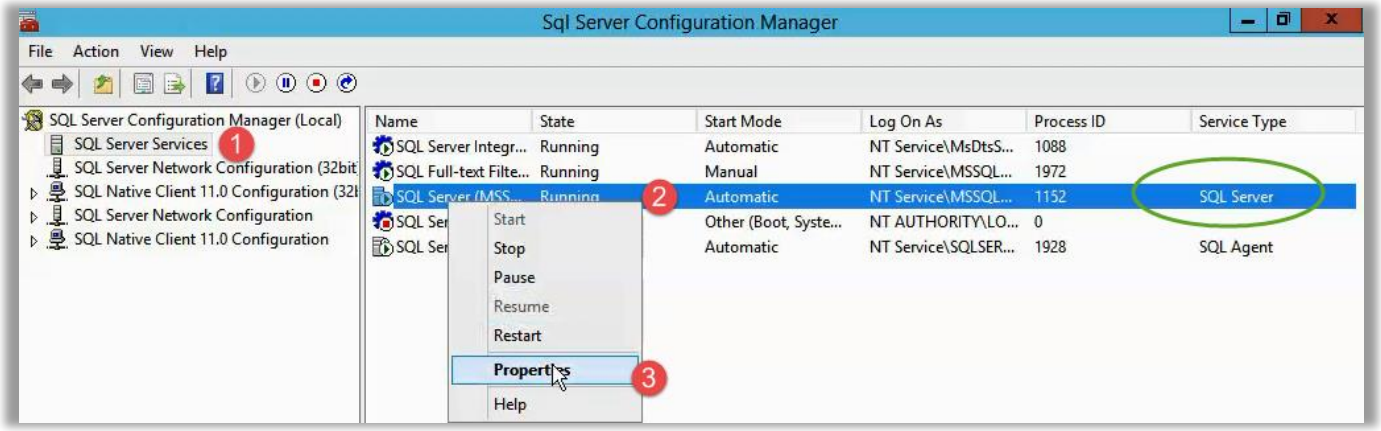


Figure 21 - Open properties of the SQL Server service

3. Click on the **AlwaysOn High Availability** tab and check the box: "**Enable AlwaysOn Availability Groups**". Press **OK** when complete.

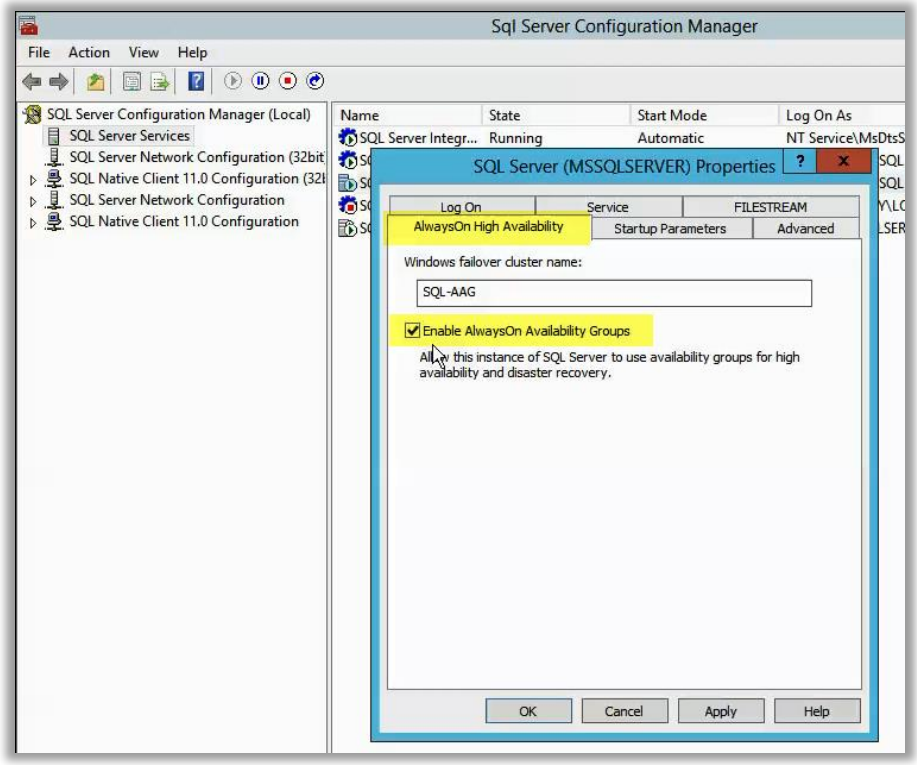


Figure 22 - Properties of the SQL Server service - Enable AlwaysOn Availability Groups

4. After enabling AlwaysOn, **Restart** the SQL Server service.

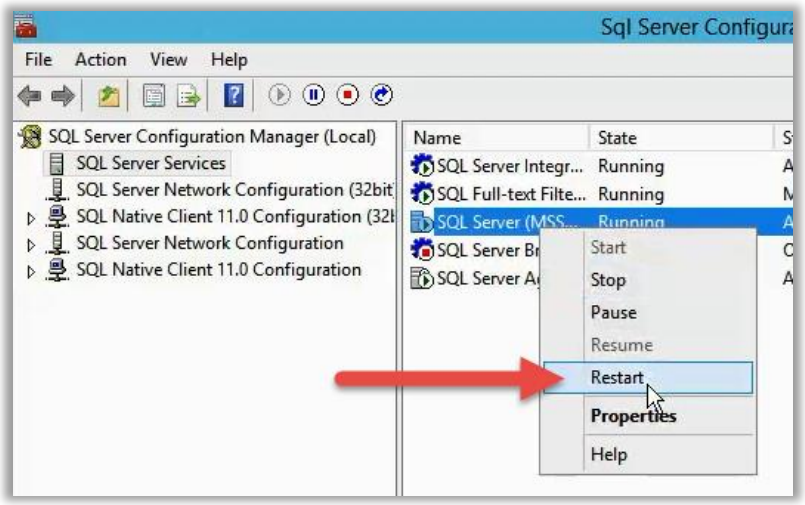


Figure 23 - Restarting the SQL Server service from within Configuration Manager

5. Repeat for each SQL Server to be clustered.

Create SQL Server Availability Group

Once SQL Server services have been successfully restarted with AlwaysOn High Availability enabled, it's time to create a SQL Server Availability Group. The steps will walk you through this:

1. Open SQL Server Management Studio and a new AlwaysOn High Availability node should be available.
2. **Right-click** on the new node and choose **"New Availability Group Wizard..."**:

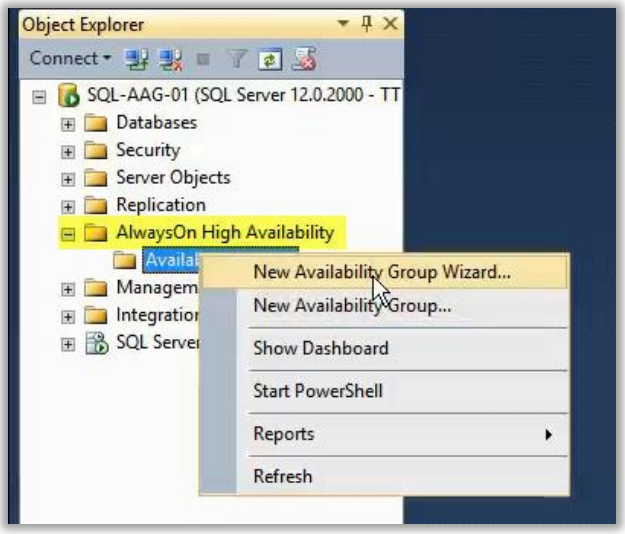


Figure 24 - Launch the New Availability Group Wizard

3. The Wizard launches and presents an overview of the process. Click **Next**

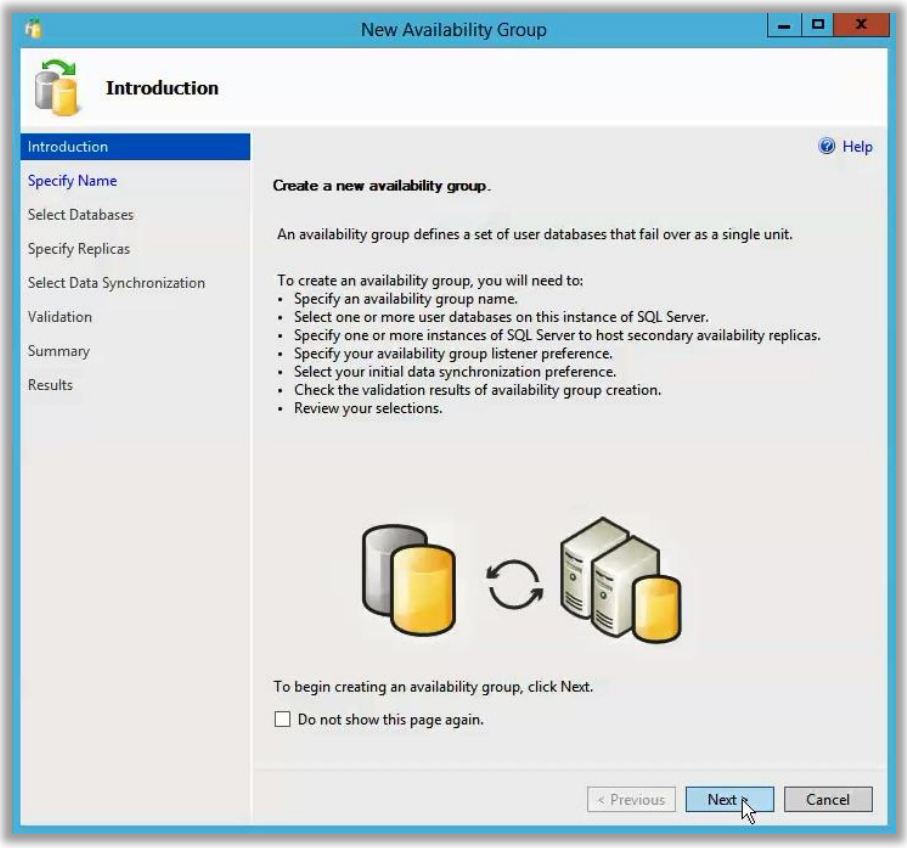


Figure 25 - New Availability Group Wizard – Overview

4. Specify a name for the Availability Group and click **Next**

Figure 26 - Specify a name

5. **Select the database(s)** to be added to the new AG and click **Next**. *NOTE: Additional databases can be added later*

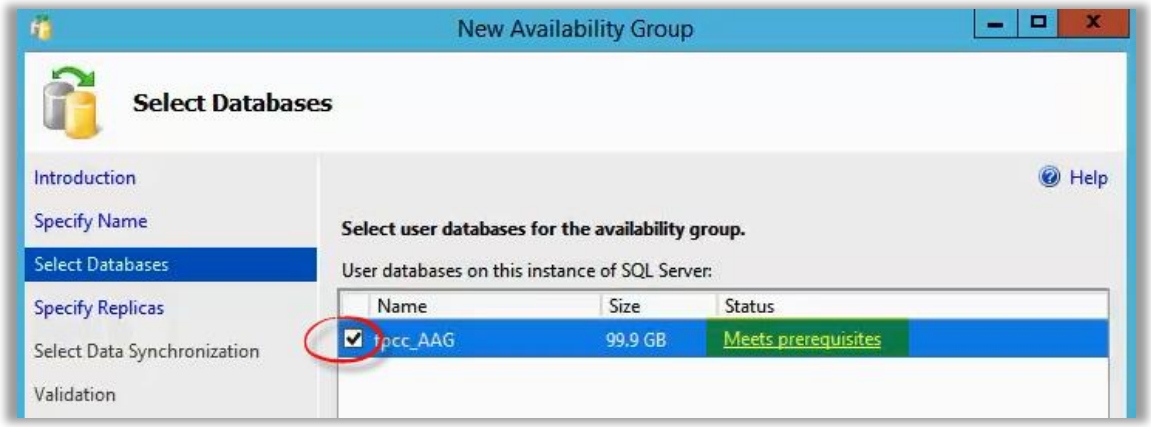


Figure 27 - Select Databases to add to the Availability Group

6. Specify Replicas. Click **Add Replica** and select one or more additional SQL Server instances to join the AG:

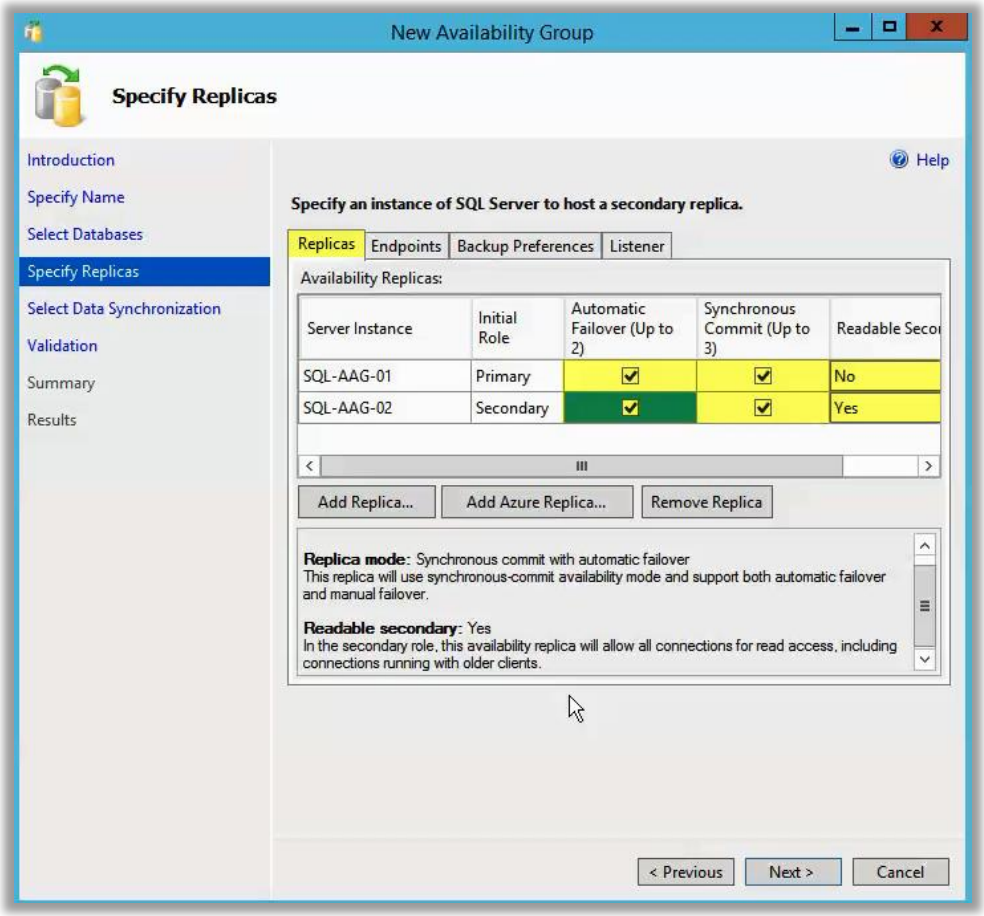


Figure 28 - Additional Replica options

7. Configure Automatic Failover, commit, and Readable Secondary options. More information can be found in Microsoft's Overview of AlwaysOn Availability Groups. If you are unsure which options to choose, configure it as shown above. All options can be adjusted later.
8. Before clicking Next, review the additional tabs for Endpoints

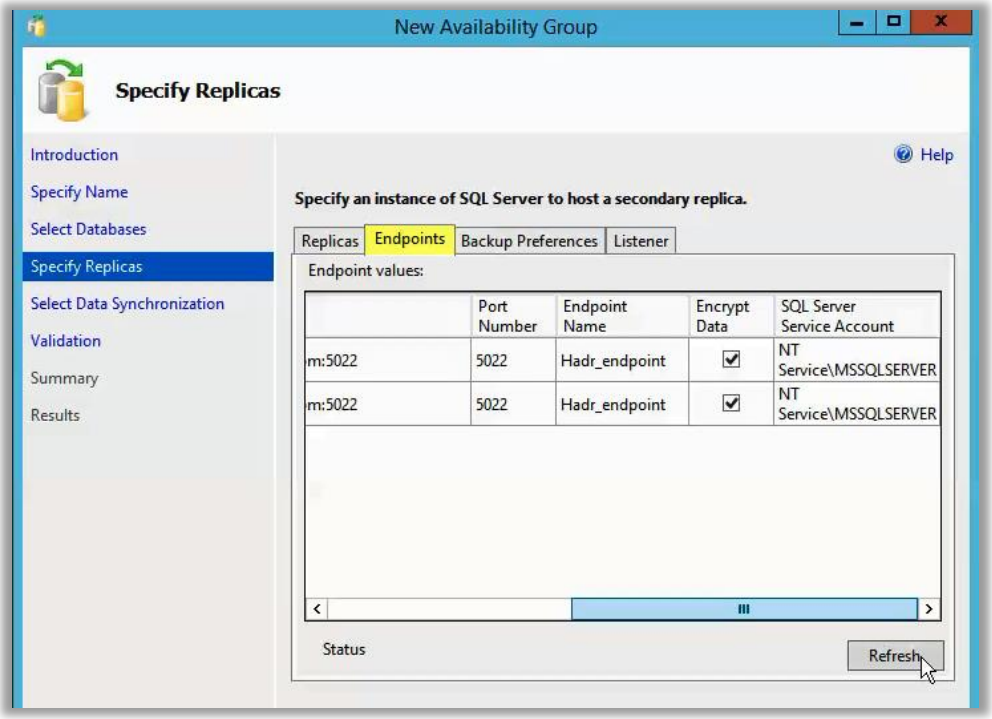


Figure 29 - Endpoint options for Replicas

9. Backup Preferences

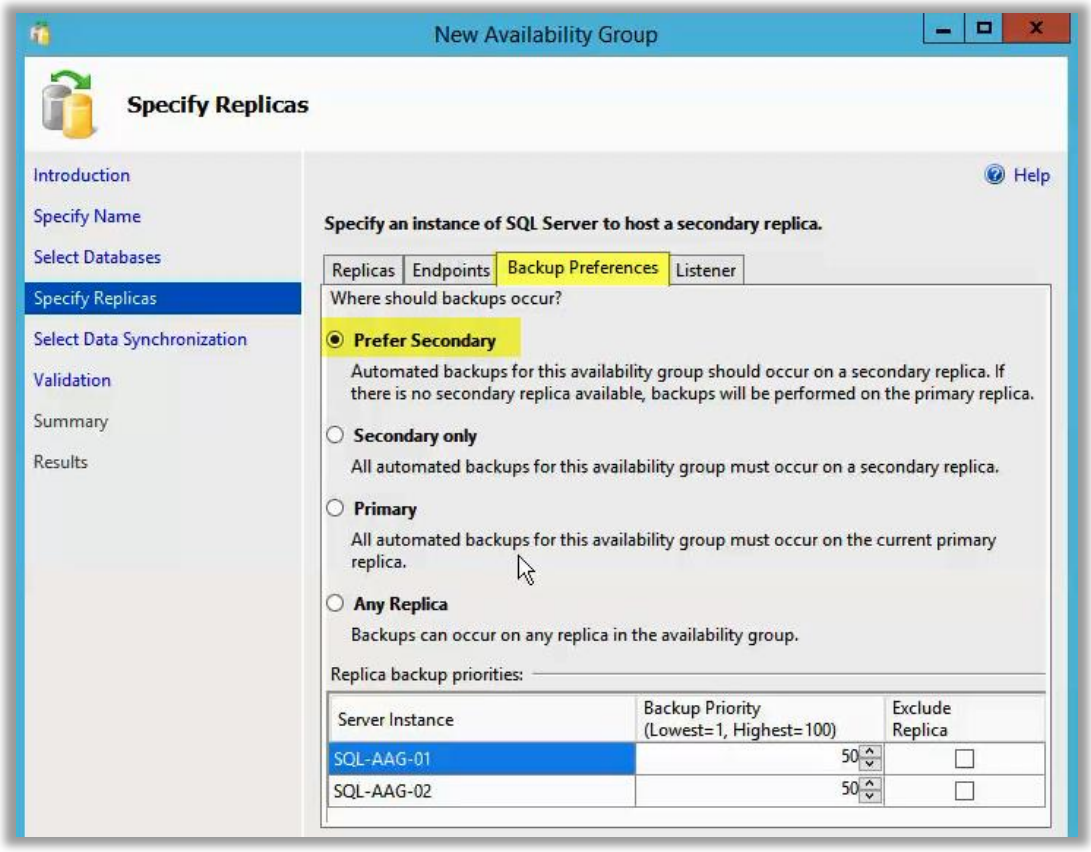


Figure 30 - Backup Preferences for Availability Group

10. Listener – Leave blank, or add a Listener (optional). More information regarding Availability Group Listeners is covered later in this guide.

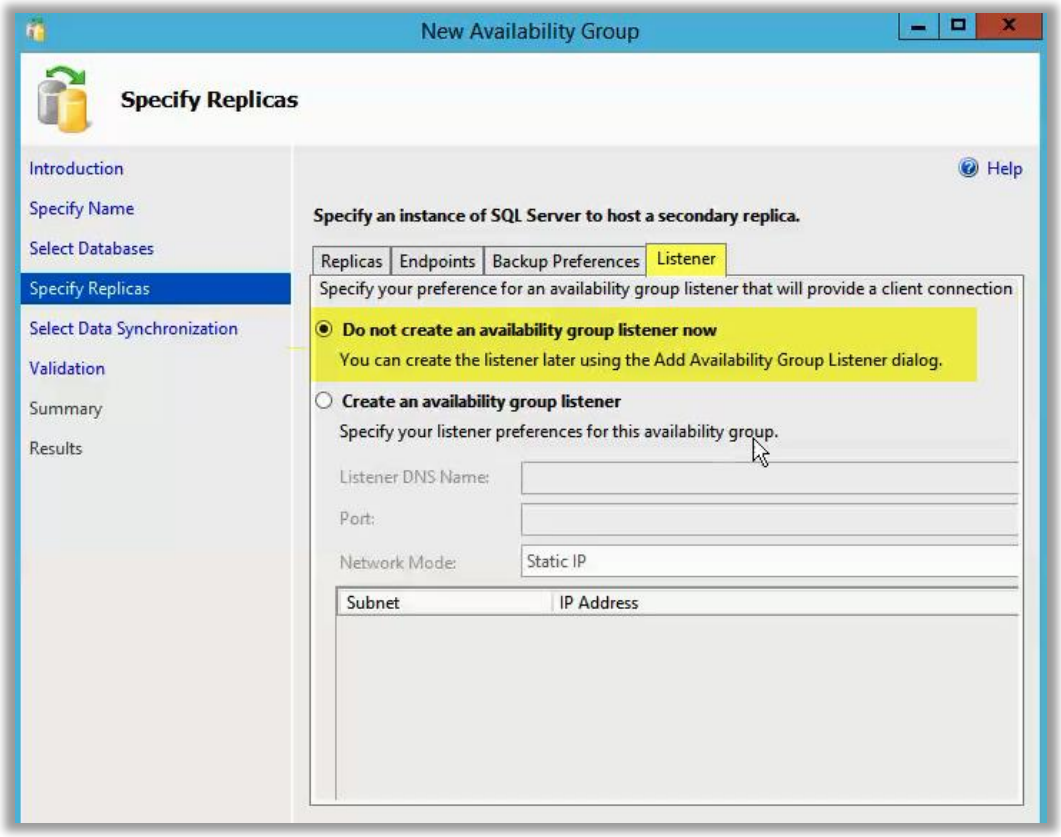
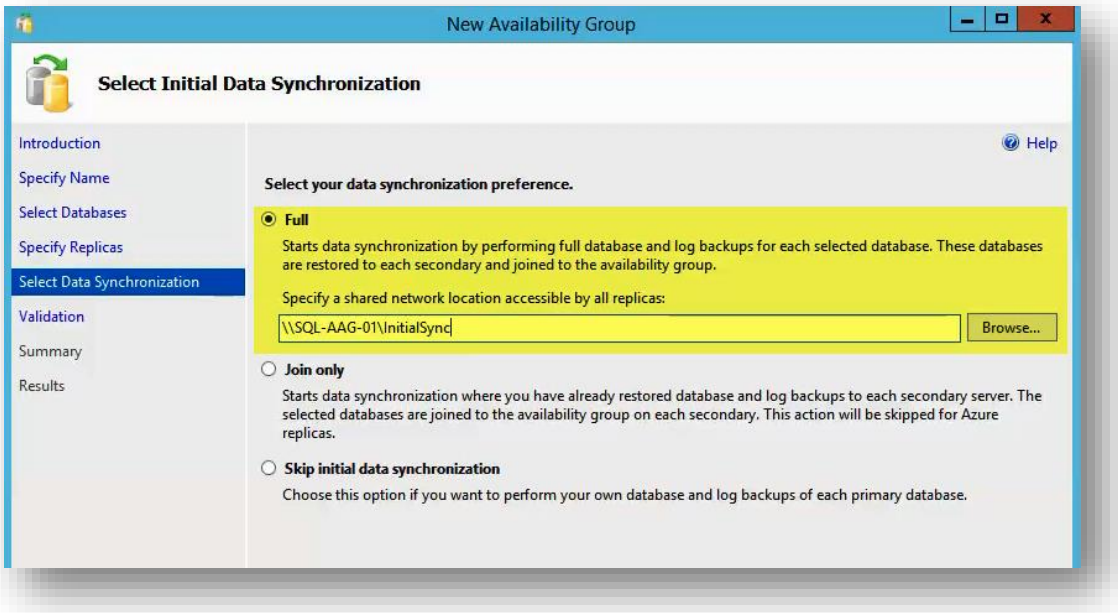


Figure 31 - Listener options for Availability Group

11. Once Replica options are selected, click **Next** to move on to Data Synchronization options:



12. Choose **Full** and provide a network share that a .bak backup file will be created in. In this example, a new folder was created on the primary SQL server's **vDisk assigned to SQL backups**, was shared, and permissions were assigned to the **DOMAIN\SQL-AAG-Computers** security group created earlier. ***NOTE: REQUIRES SUFFICIENT FREE SPACE FOR A FULL BACKUP!***
13. Confirm that Validation Checks are OK

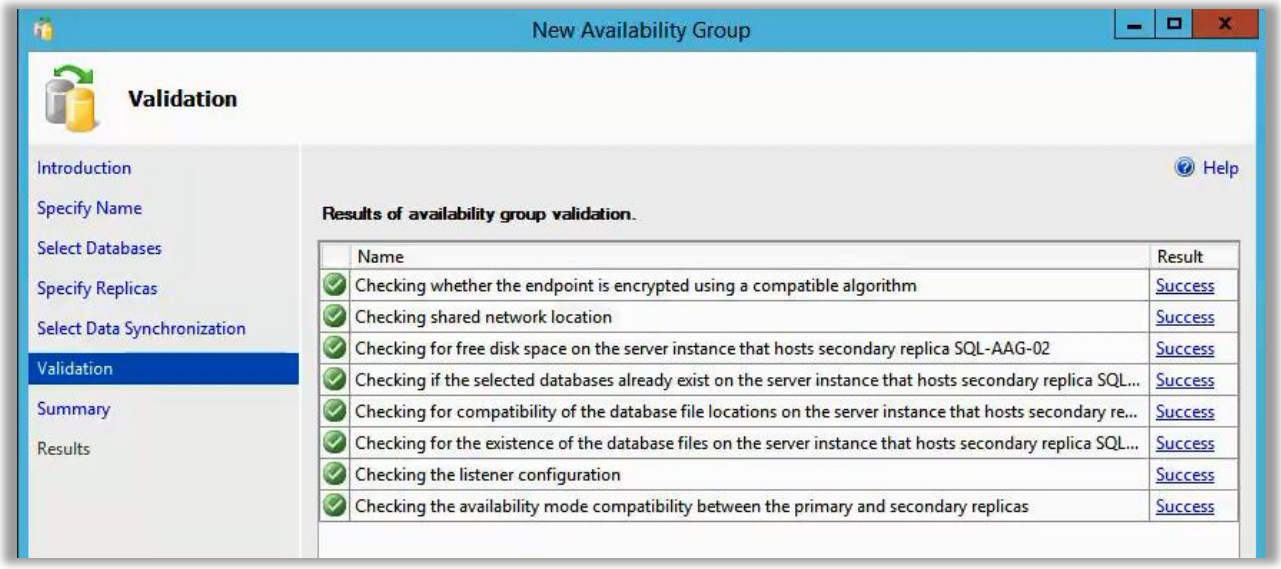


Figure 32 - Validation Checks – review any line times that do not have a Successful result

14. After validation is complete, review the summary, and click **Finish**.

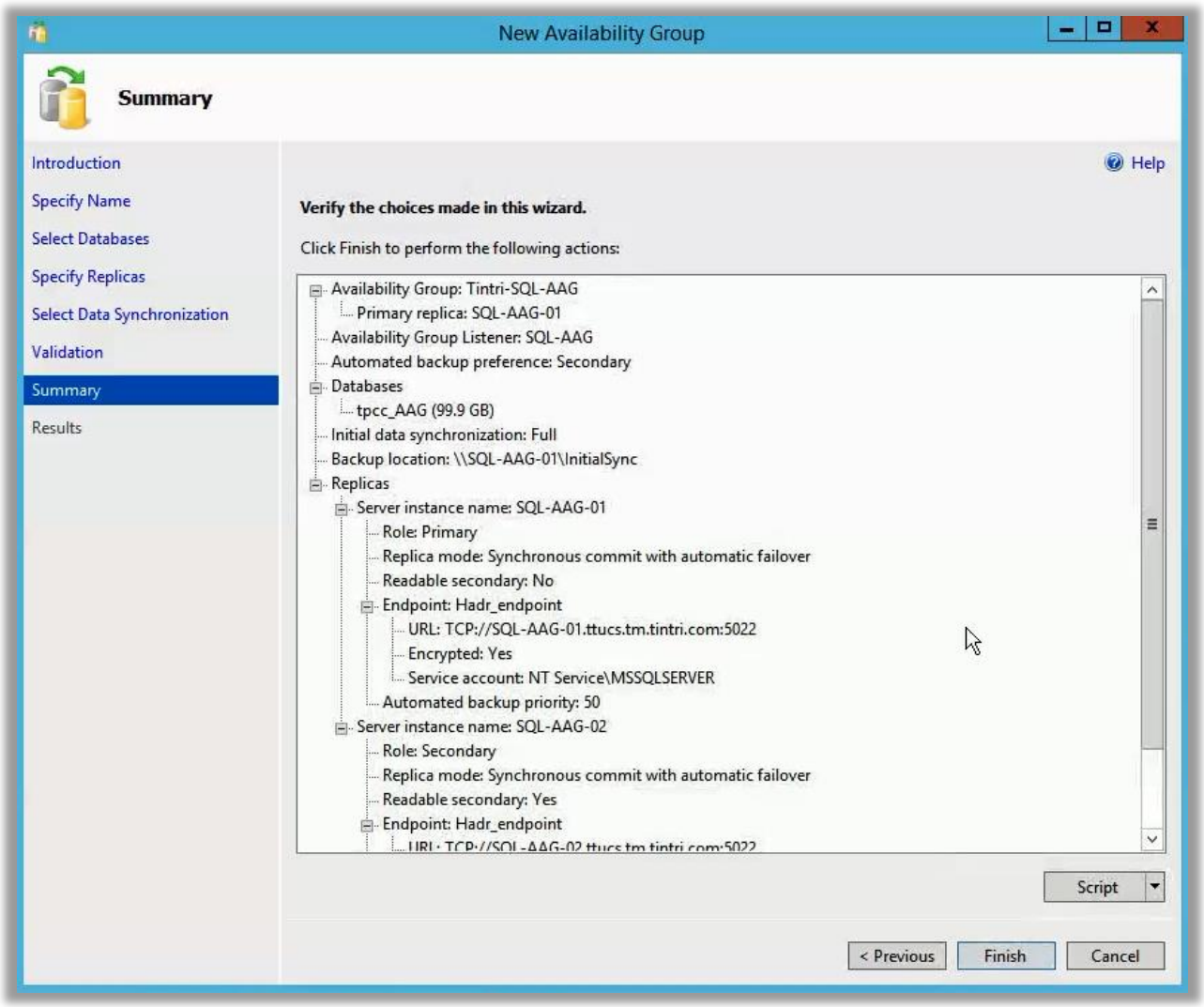


Figure 33 – Summary of options to be used to create the Availability Group

15. Once the process starts, click on **More Details** to see an itemized list of each step, as well as the progress of each:

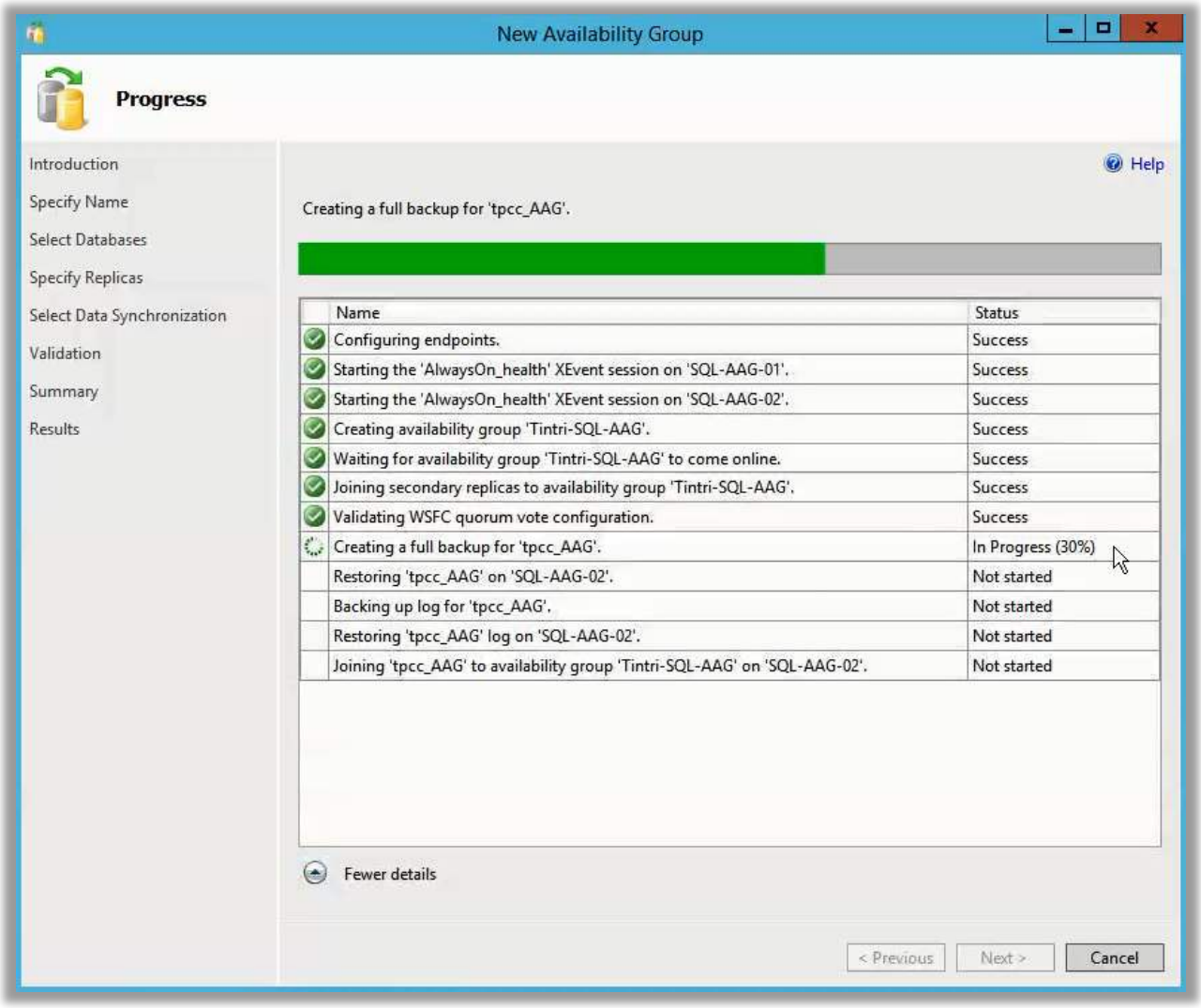


Figure 34 - Availability Group Wizard - Progress of final tasks

If the wizard fails to join the availability group, for any reason, review the Error using the link provided in the Result column. Often the errors are well described, but in other cases the root cause may be more obscure and harder to identify. Review the Windows System, Security, Application and SQL event logs for clues and double-check that permissions were assigned as previously outlined.

Adding Databases to the SQL Server Availability Group

Adding additional databases to an existing SQL Server Availability Group is fairly straight-forward and can be performed while the database is online, without interruption. There may be some overhead associated with some of the operations, primarily seen with backing up the database, transferring it to the other server, and restoring it.

Refer to Appendix C for detailed Step-by-Step instructions.

HammerDB was used to generate load using the schema creation process within a newly-created single instance database. The new database was actively being written to under heavier load than typically found in day to day operations of most environments. In this case, the new database was being populated at a rate of over 2 Million transactions per minute (tpm). Although the tpm rate dropped by ~15% during process of making the DB highly available, none of the HammerDB client nodes lost their connections to the database and they continued populated the database with unique, random data without interruption.

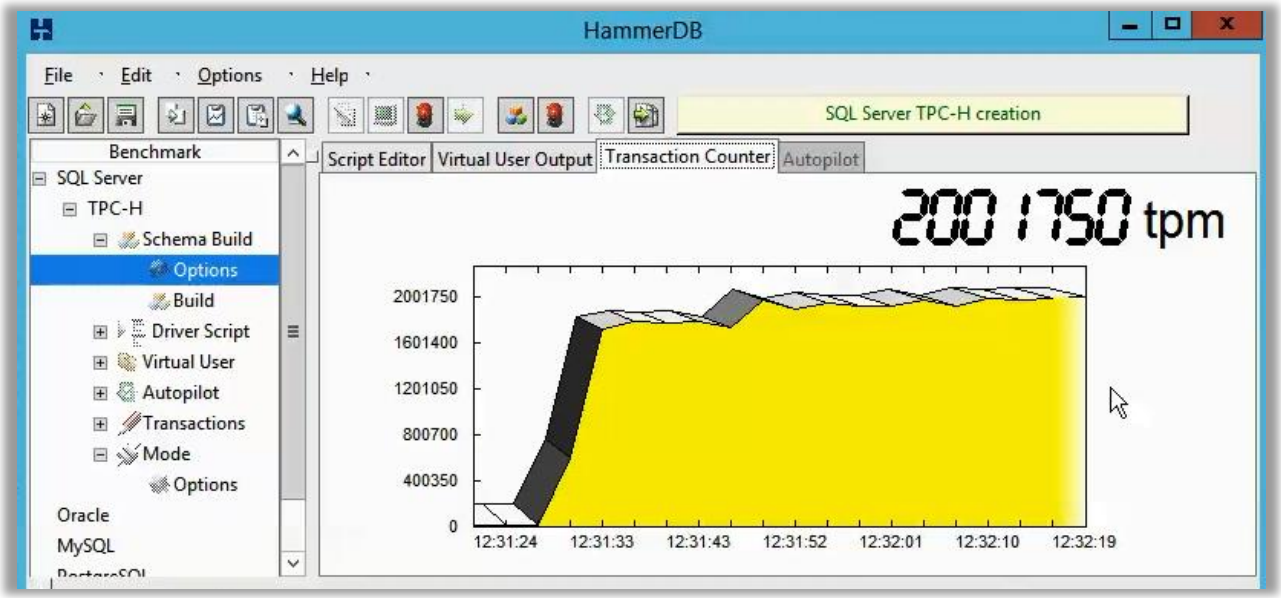


Figure 35 - HammerDB was used to apply load while testing the AG creation process and various AG failover/resiliency tests

To add a database to an existing AG, open SQL Management Studio, **right-click on Availability Databases**, and select **Add Database…**, as shown here:

Refer to Appendix C for detailed Step-by-Step guidance.

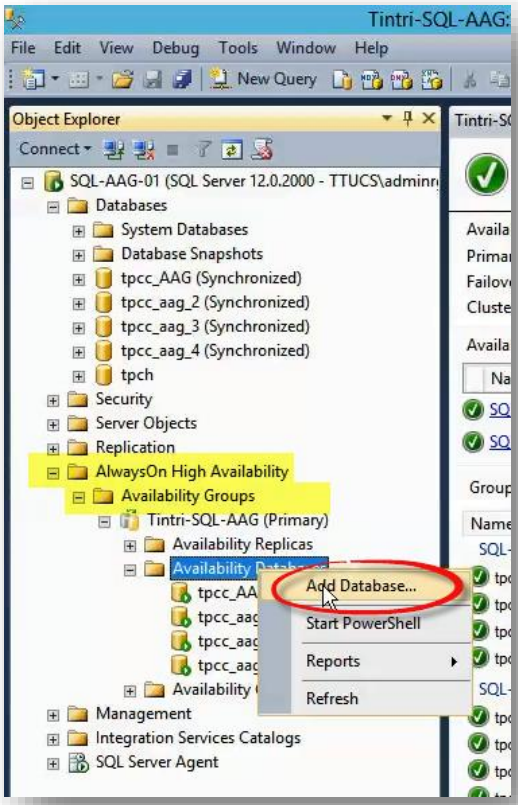


Figure 36 – Right-click the Availability Databases node and Select "Add Database…" to invoke launch the wizard

With databases added to the AG, use the Availability Dashboard to verify that all server nodes are healthy, and that the AG itself has a healthy status, as shown here:

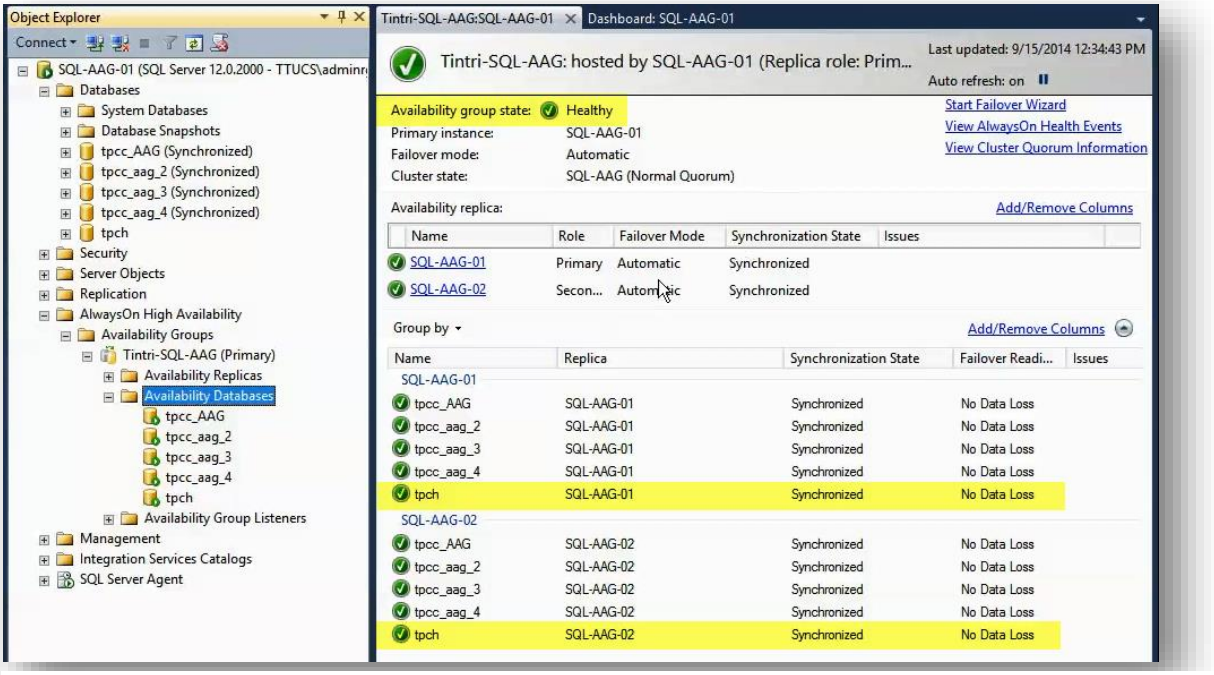


Figure 37 - Green is good! Use the dashboard to check the health status of your AlwaysOn Servers and Databases. Right-click AlwaysOn Availability and choose "Show Dashboard"

Create an Availability Group Listener (Optional)

An availability group listener is a virtual network name (VNN) to which clients can connect in order to access a database in a primary or secondary replica of an AlwaysOn availability group. An Availability Group Listener is assigned a unique DNS name and one or more IP addresses.

While availability group listeners enable support for failover redirection and read-only routing, **client connections are not required to use them**. A client connection can also directly reference the instance of SQL Server instead of connecting to the availability group listener. In this guide, we used the Virtual IP and unique DNS name assigned to the windows failover cluster for client database connections instead of an availability group listener.

There are no unique requirements for Availability Group Listeners with respect to running your SQL Server Availability Group databases on Tintri VMstores, so we've left this step as optional, but felt it worth mentioning for completeness. For more information, refer to the following two links:

- Availability Group Listeners, Client Connectivity, and Application Failover
- Create or Configure an Availability Group Listener

VMware vCenter Settings

DRS Cluster Settings

Create a rule to prevent SQL Server VMs within the same Availability Group from running on the same host. Running two or more servers responsible for keeping a single logical database highly available creates a single point of failure if the host fails affecting both the SQL Server VMs. Implement DRS rules for your highly-available VMs to provide better resiliency to complete host failures.

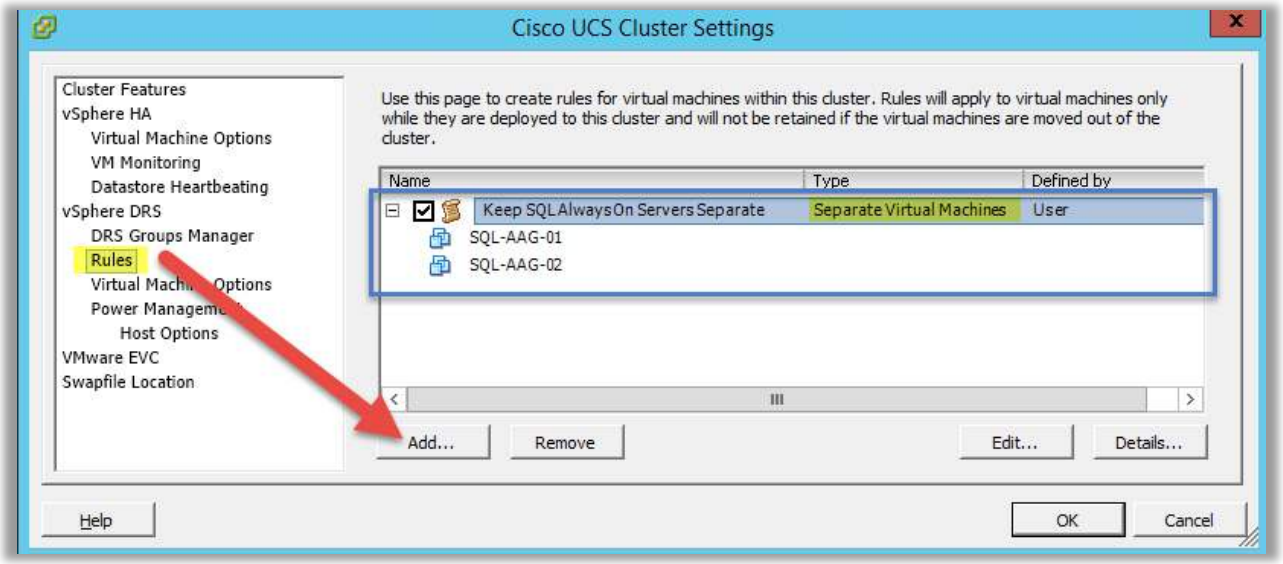


Figure 38 - DRS Rule that prevent Clustered SQL Database VMs from sharing the same host

---

***DO**: Create a rule to prevent SQL Server VMs that are members of the same Availability Group (AG) from running on the same host.*

Conclusion

SQL Server AlwaysOn Availability Groups allow us to obtain a higher level of application availability than is otherwise possible in a stand-alone instance of a SQL Server. This guide has provided information around SQL Server Availability Groups and how to configure in a supported configuration on Tintri VMstore storage appliances.

Microsoft Clustering Services (MSCS) has historically been difficult to configure within a virtual environment, but was required in order to cluster SQL Server. Thanks to Microsoft technology advancements and the advent of Windows Failover Clustering (WSFC) and SQL AlwaysOn technology, MSCS is no longer required to deploy and manage highly available SQL Server databases.

Thank you for choosing Tintri. We hope the options and guidance presented in this guide enable you to leverage this new technology. For additional information about Tintri VMstores and other technical whitepapers and resources, visit **www.tintri.com.**

References

Tintri Links

- Microsoft SQL Server Best Practices Guide on Tintri

Microsoft Links

- SQL Server Failover Cluster Installation
- Windows Server 2012 R2 – Failover Clustering Overview
- Overview of AlwaysOn Availability Groups
- Windows Server Failover Clustering (WSFC) with SQL Server
- Failover Cluster Permissions (Windows 2012 R2)
- Deploy an Active Directory-Detached Cluster
- WSFC Quorum Modes and Voting Configuration
- Prerequisites, Restrictions, and Recommendations for AlwaysOn Availability Groups
- Setup Login Accounts for Database Mirroring or AlwaysOn Availability Groups
- Availability Group Listeners, Client Connectivity, and Application Failover
- Create or Configure an Availability Group Listener
- SQL 2014 AlwaysOn Enhancements (compared to SQL 2012)
- Features Supported by the Editions of SQL Server 2014
- Cluster-Aware Updating Overview

Other Links

- HammerDB – Testing Tool

Appendix A – Step-by-Step: Windows Failover Cluster Creation

Follow the steps below to validate and create a Windows Failover Cluster:

1. Open the Failover Cluster Manager, which can be accessed from the Tools menu of the main Server Manager dashboard in Windows 2012:

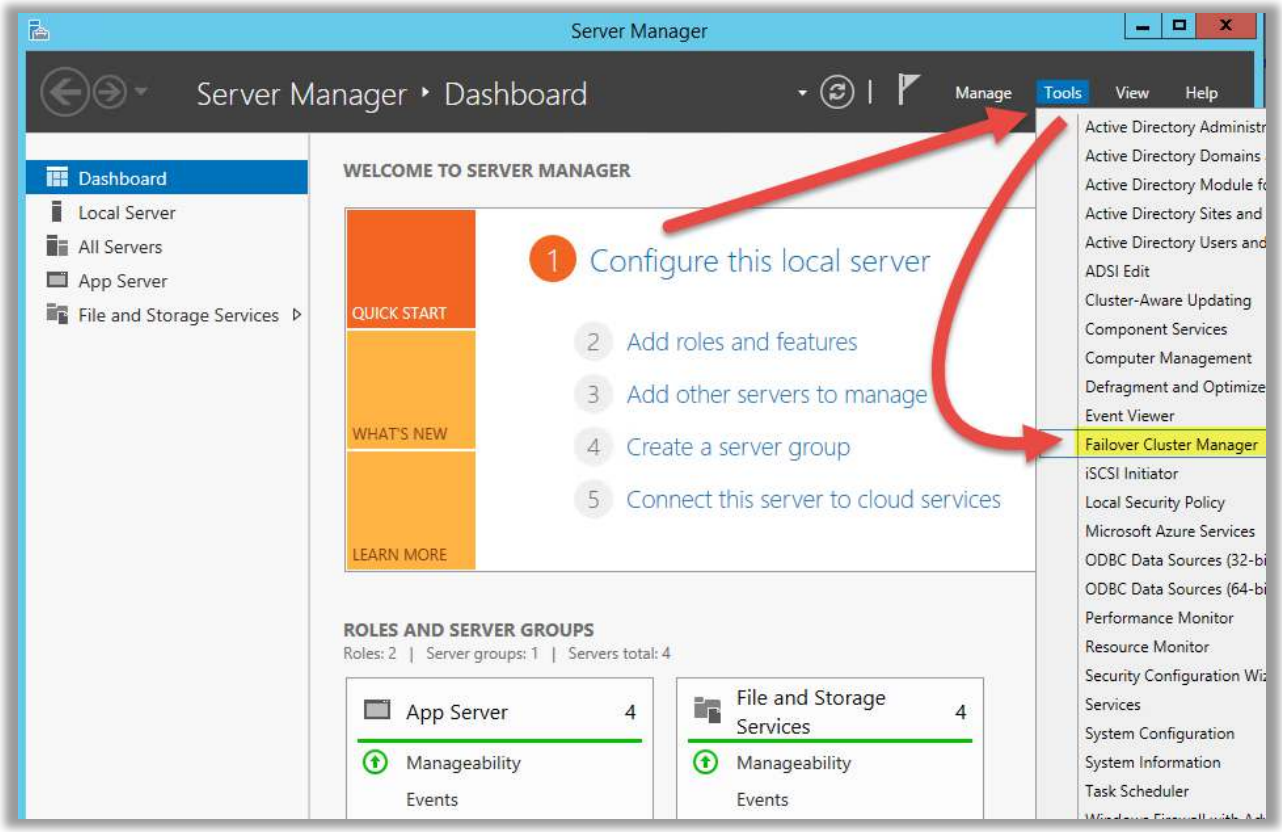


Figure 39 - Accessing the Failover Cluster Manager

2. Click on Validate Configuration:

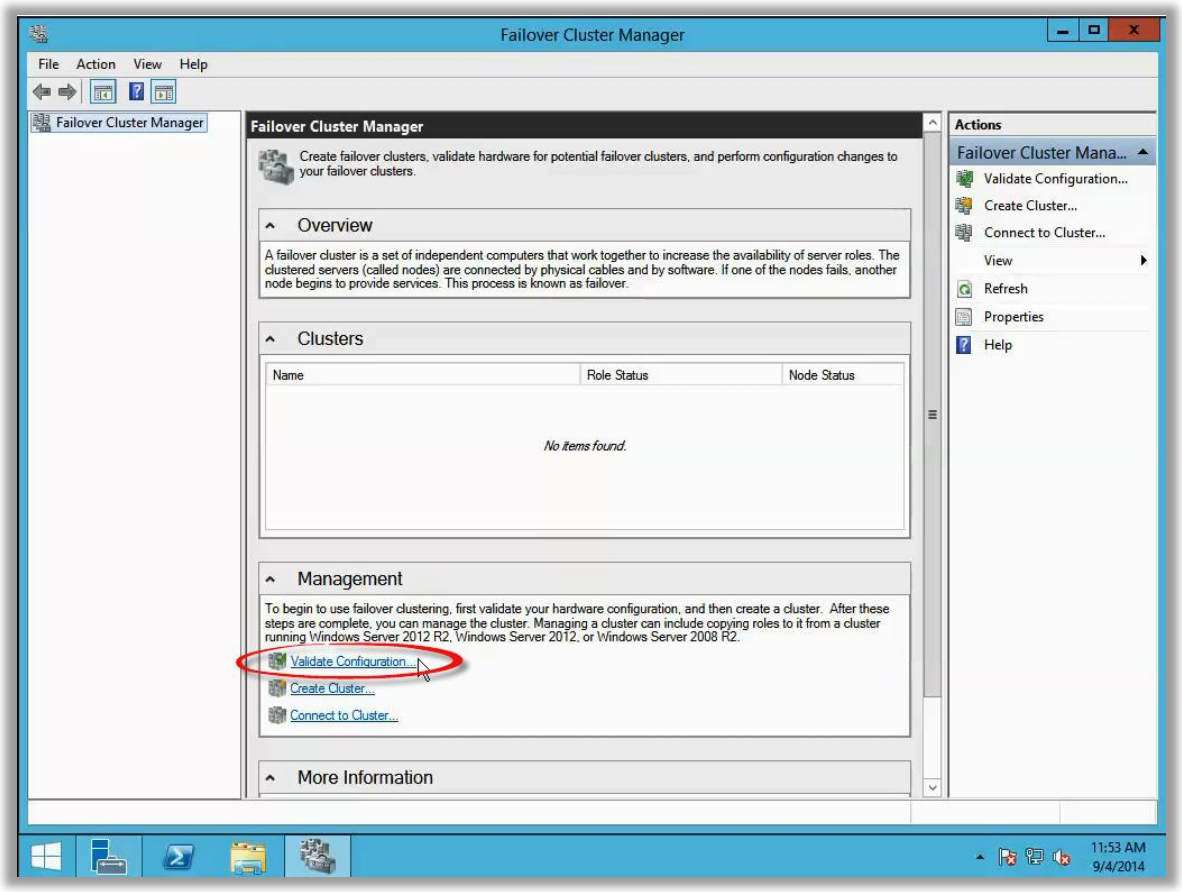


Figure 40 - Click on Validate Configuration

3. Click Next to proceed beyond the Intro:

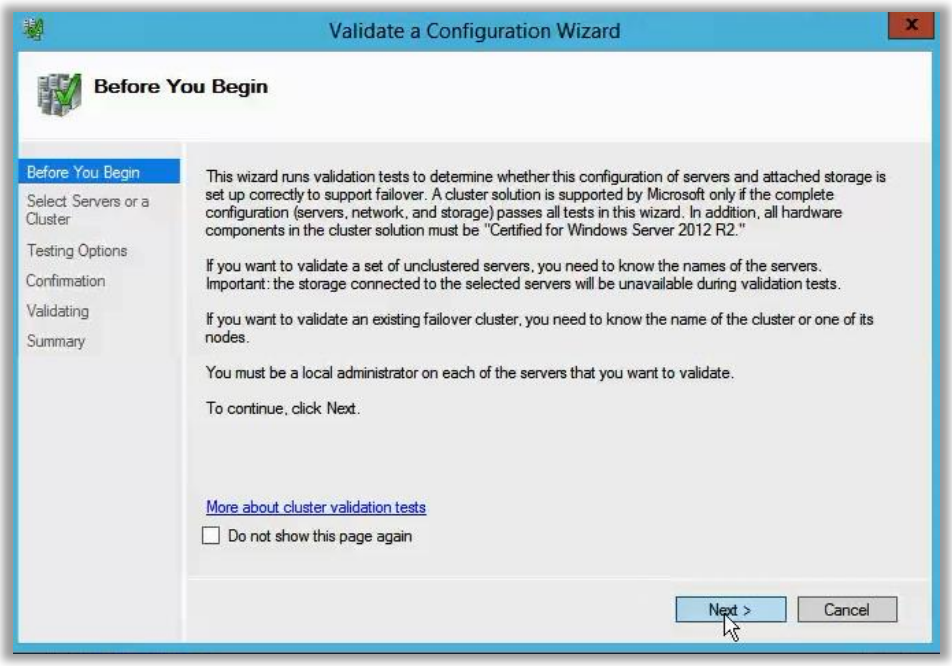


Figure 41 - Validate a Configuration Wizard - Click Next to proceed

4. Select servers to add to the cluster:

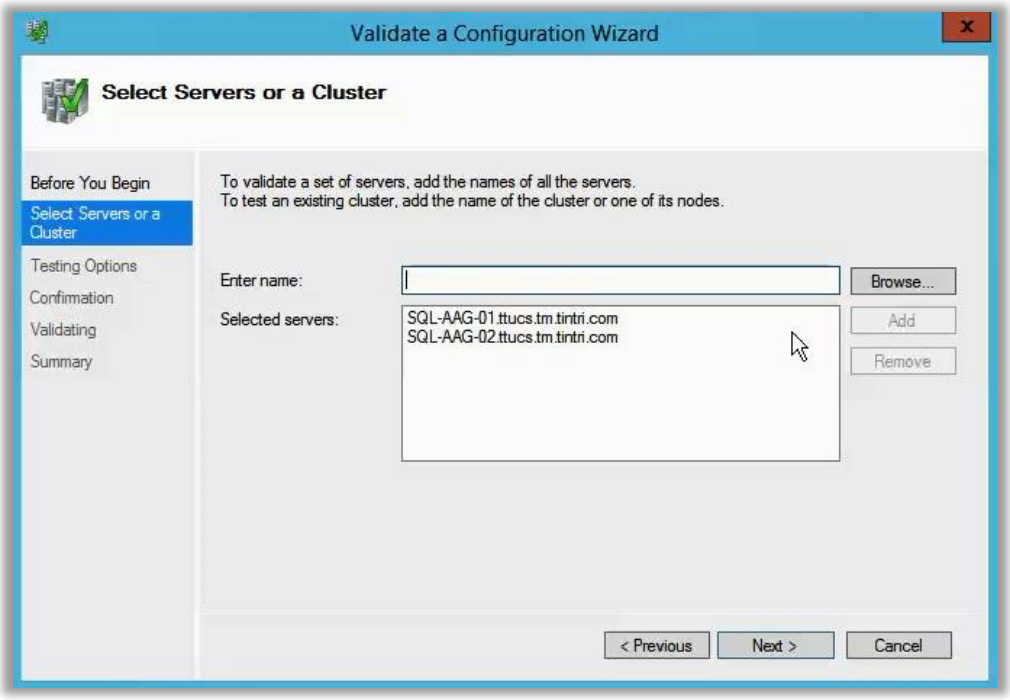


Figure 42 - Select Servers you want to add to a cluster using the Browse... button, the click Next

5. Choose “Run all tests” and click Next:

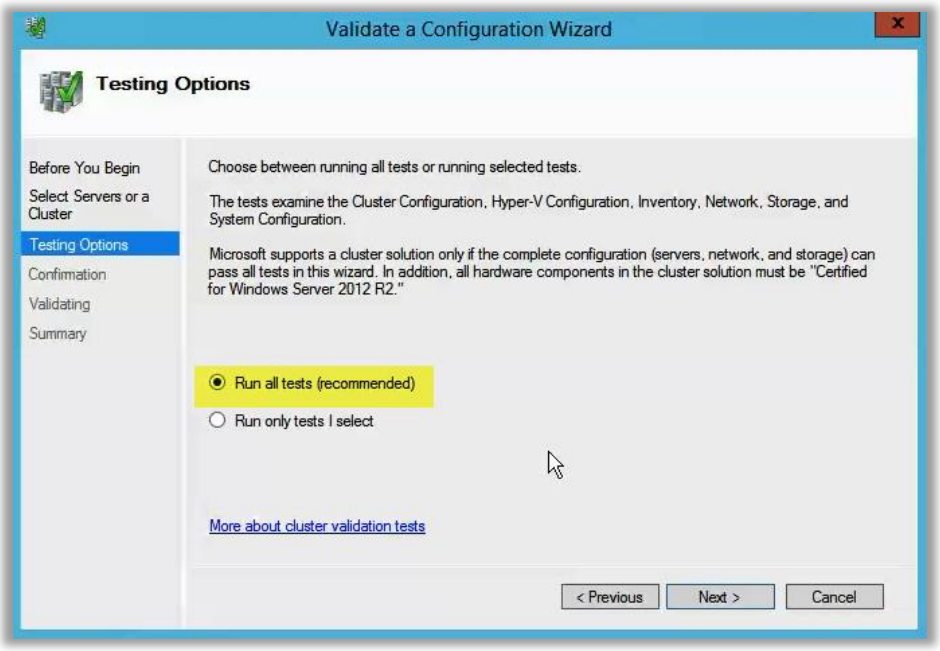


Figure 43 - Run all tests and click Next

6. Wait for Validation tests to run:

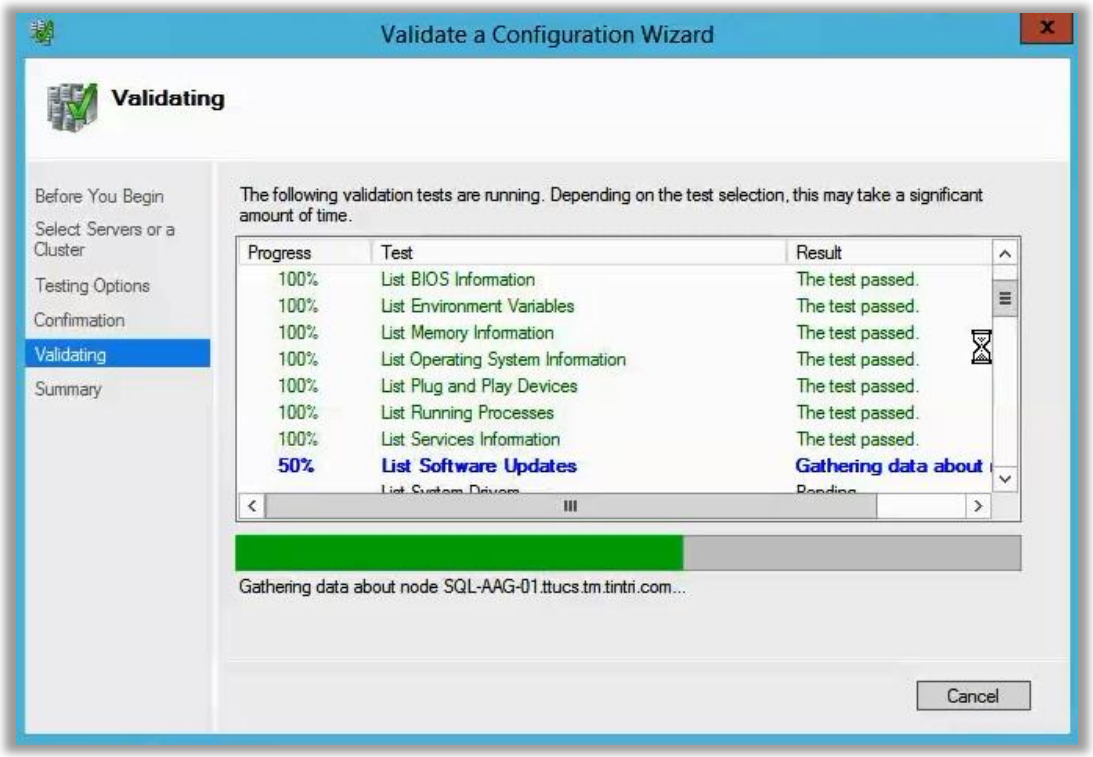


Figure 44 - Validation....

7. Review the Summary:

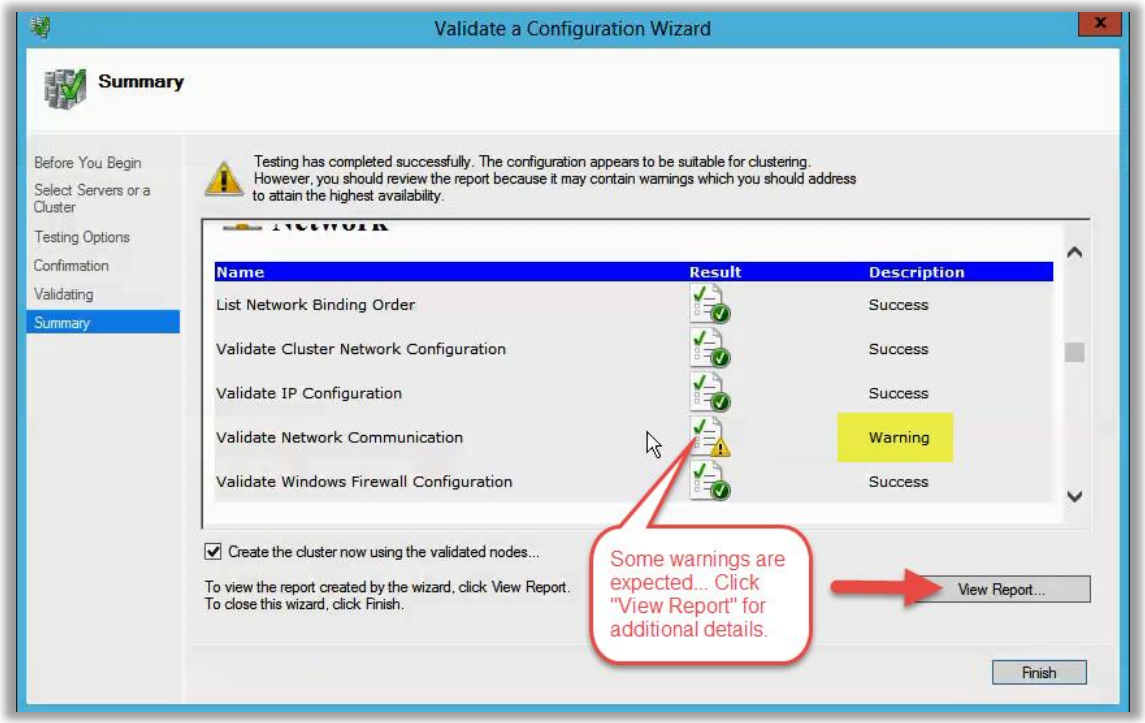


Figure 45 - Summary of Validation results

8. After reviewing the results for errors and warnings. Refer to the previous section on creating a Windows Failover Cluster for detail on some known false-positive errors and warnings.
9. **Check the checkbox** for "**Create the cluster now**..." and click **Finish** to launch the Create Cluster Wizard, as shown here:

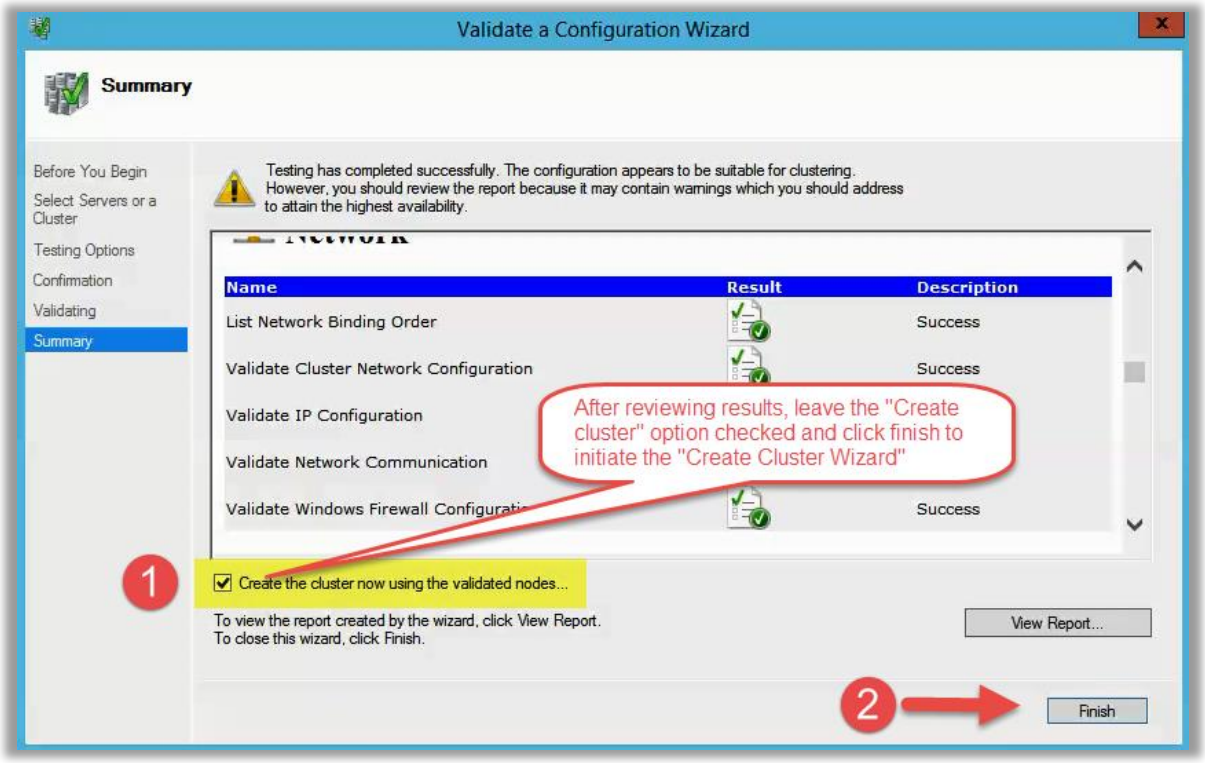


Figure 46 - Click Finish to launch the Create Cluster Wizard

10. Click **Next** to proceed past the intro:

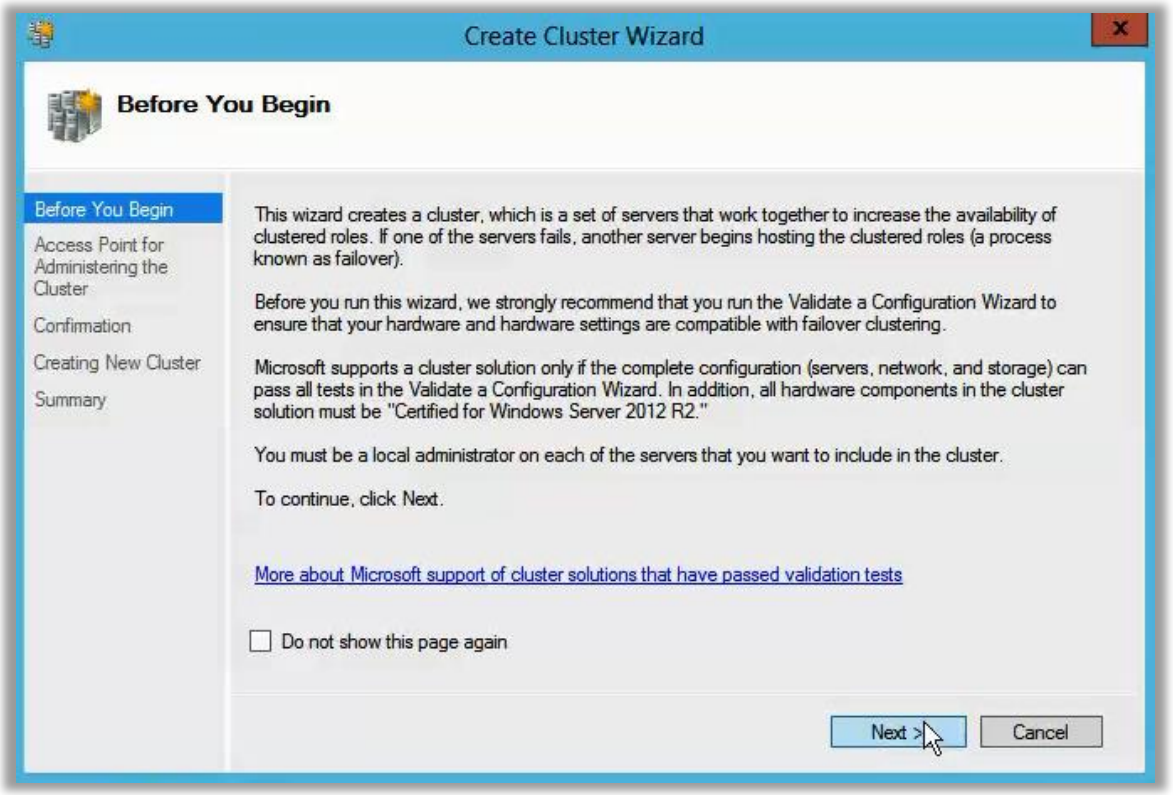


Figure 47 - Intro screen for the Create Cluster Wizard

11. Provide a name for the cluster. The name supplied here will create a Computer Object in the AD, as described in the previous configuration section:

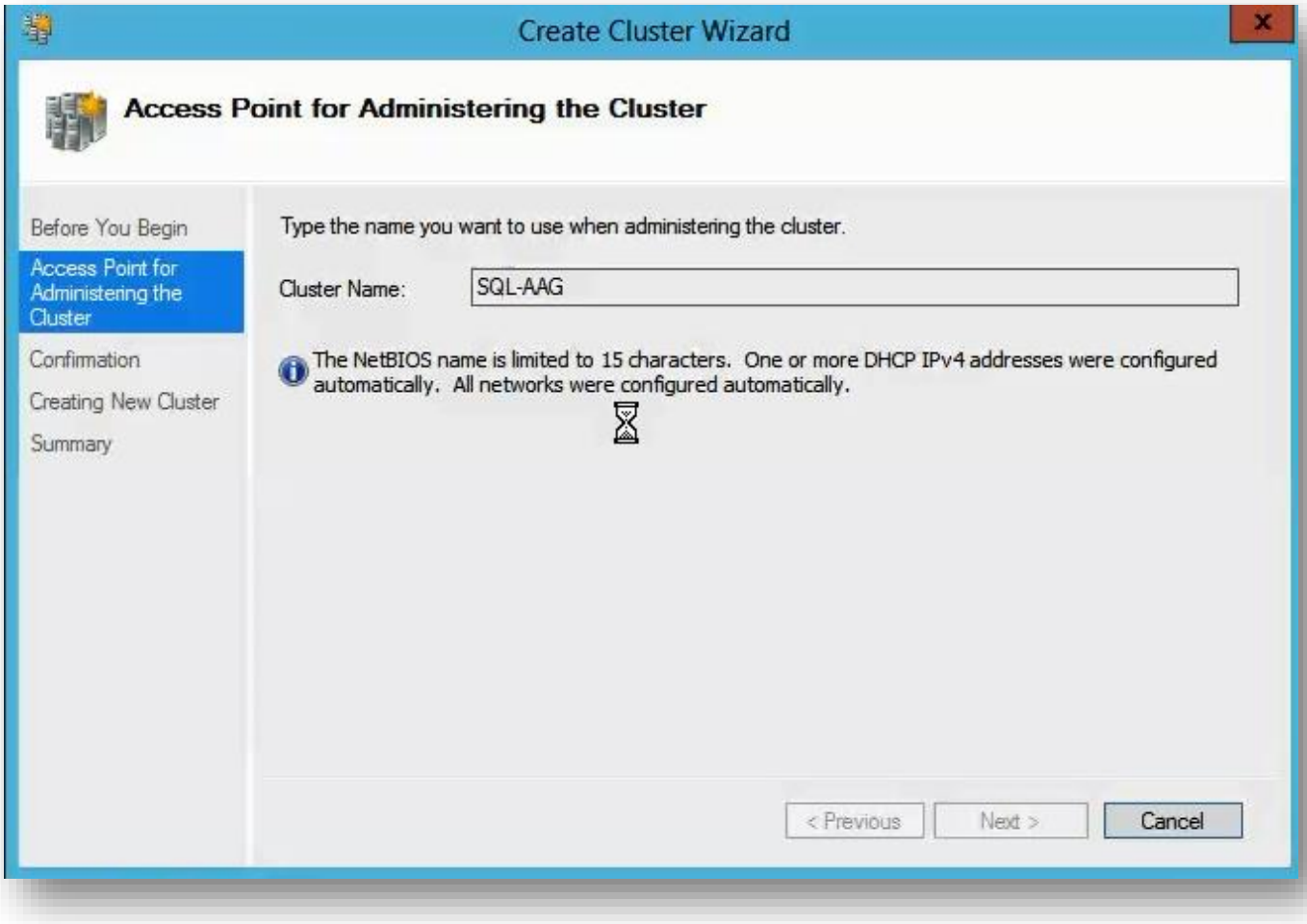


Figure 48 - Provide a name for your cluster. This will be the name of the virtual IP of the active cluster node

12. On the confirmation, **UNCHECK** the "Add all eligible storage to the cluster" checkbox and click **Next** to start the cluster creation:

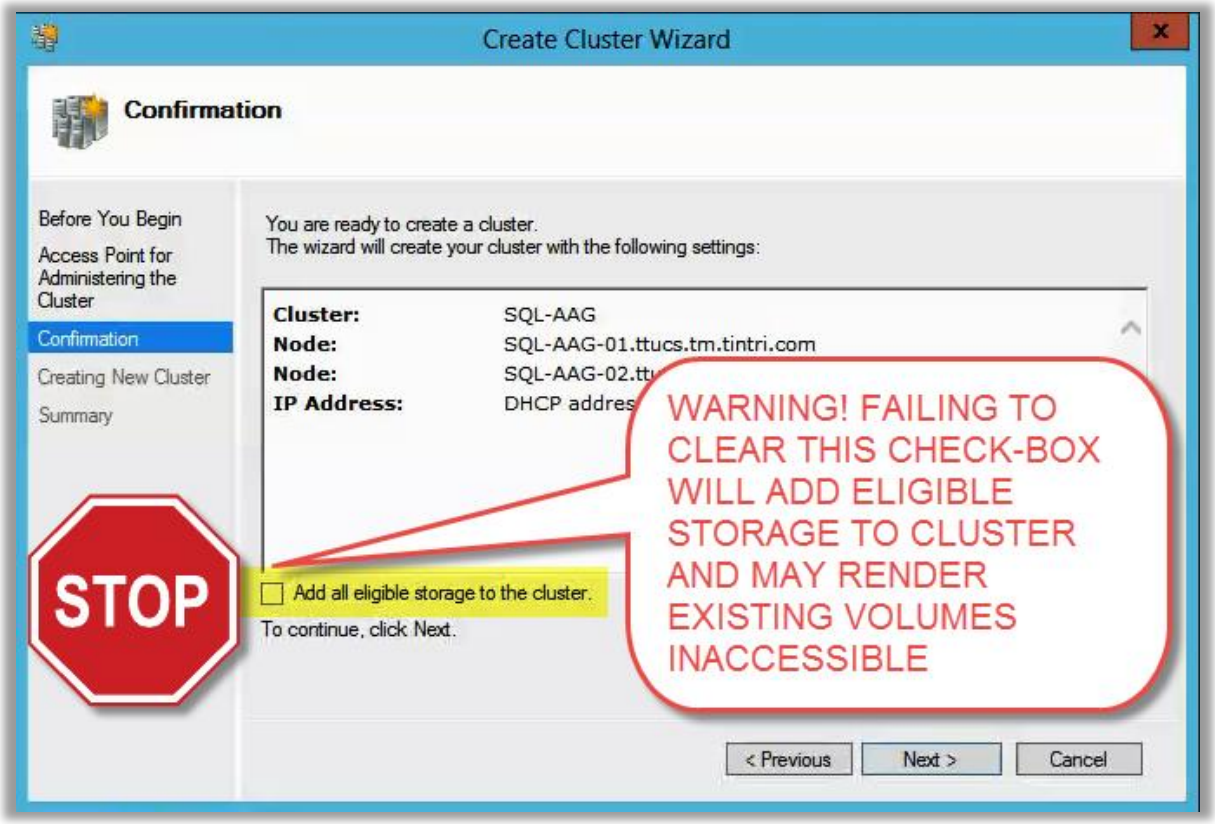


Figure 49 - On the confirmation, UNCHECK the "Add all eligible storage to the cluster" checkbox

13. Wait while the new cluster is created:

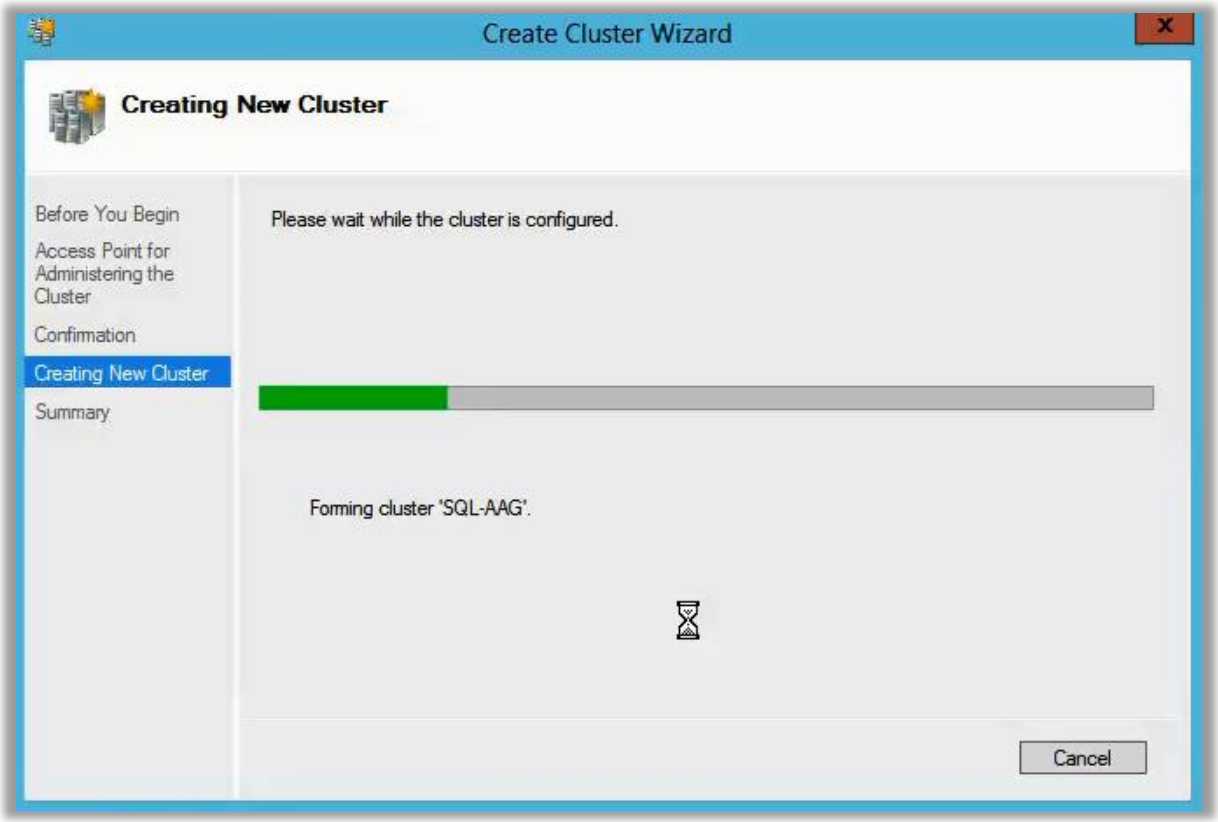


Figure 50 - A new cluster is being created

14. On the Summary screen, click **View Report** for a detailed report on the creation tasks and review for warnings and/or errors.

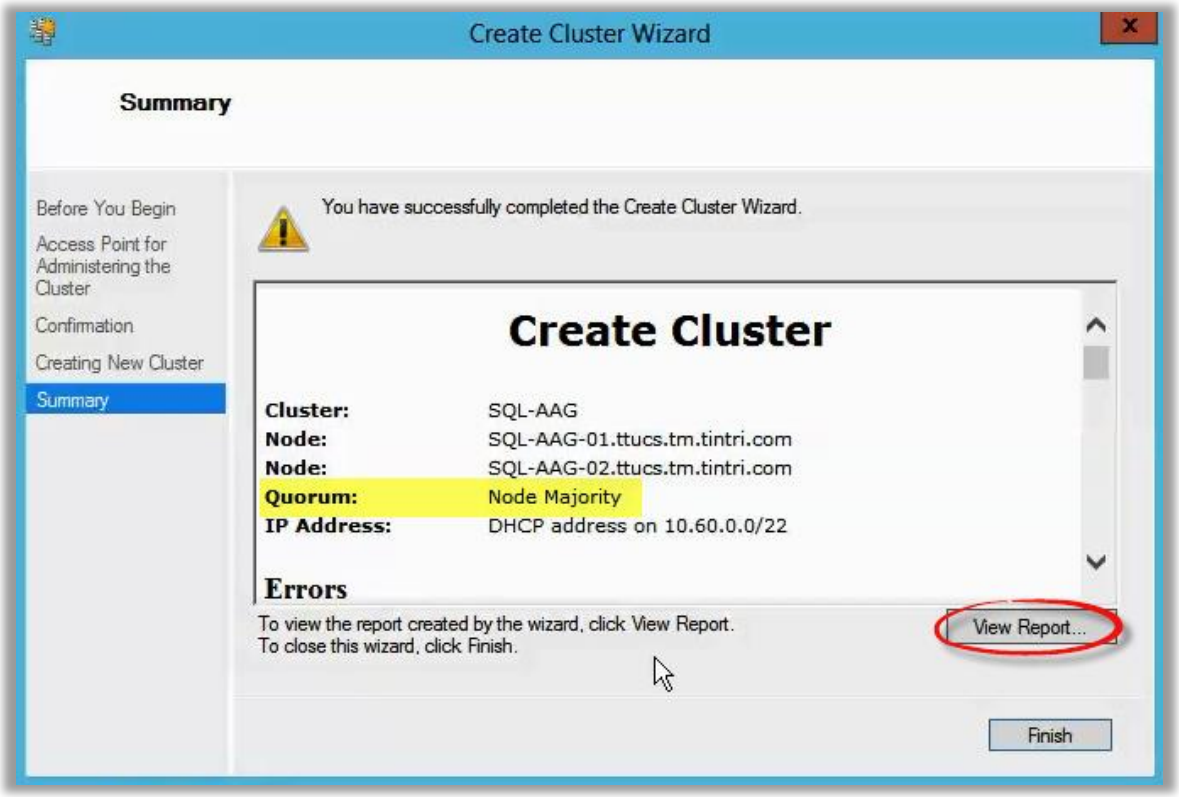


Figure 51 - Cluster Summary - Click "View Report" for details

You have now successfully created a Windows Failover Cluster!

Appendix B – Step-by-Step: Configuring Cluster Quorum Settings

To configure a File Share to be used as the cluster quorum, follow the steps below.

1. From within Failover Cluster Manager, **right-click** on the newly-created Cluster and select **More Actions – Configure Cluster Quorum Settings:**

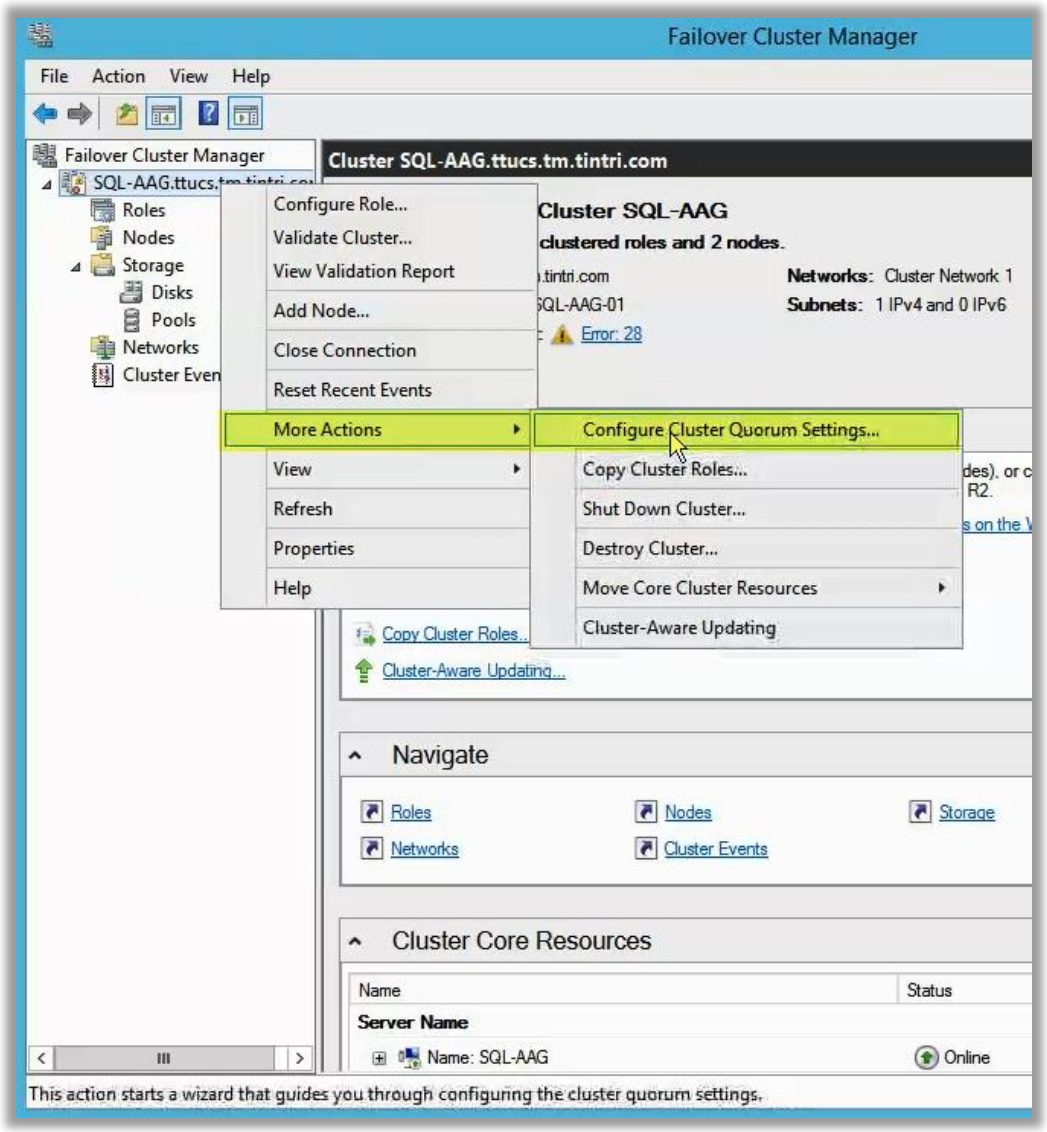


Figure 52 - Configure Cluster Quorum Settings can be found under the "More Actions" menu

2. Click **Next** to proceed past the intro:

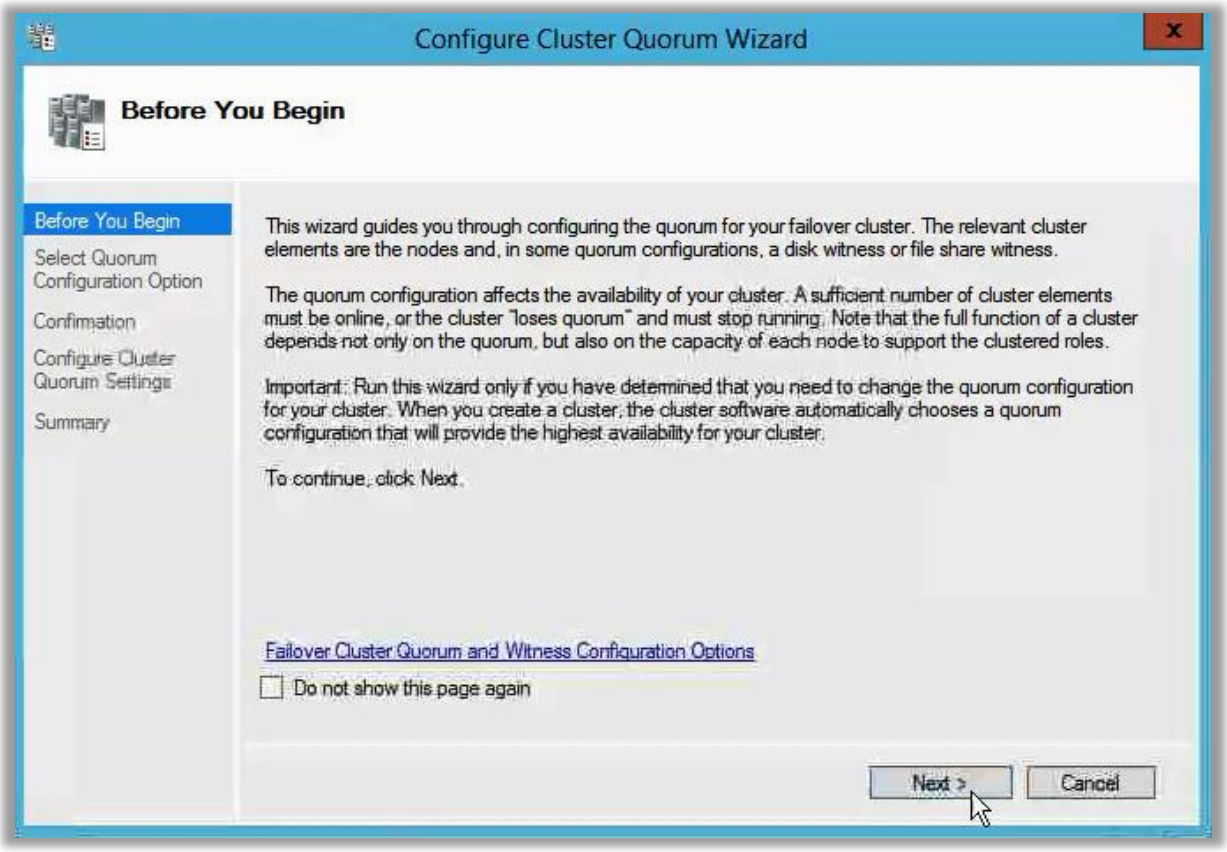


Figure 53 - Cluster Quorum Configuration Wizard - Intro

3. Choose the second option (**Select Quorum Witness**) and click **Next:**

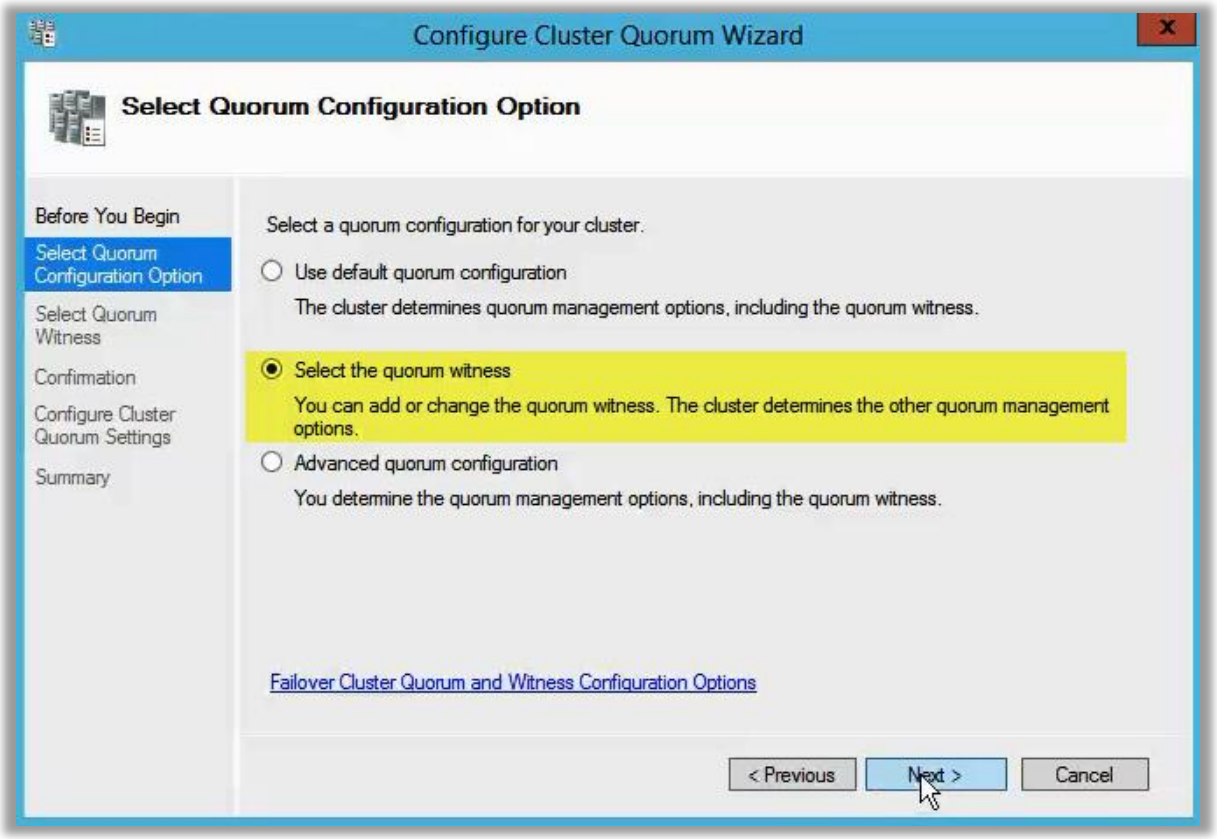


Figure 54 - Cluster Quorum Configuration Wizard - Select the Quorum witness Option

4. Provide a file share path for the quorum. This is the share that was created on a file server in the Prerequisites section. Use a FQDN for the host, if applicable, and click **Next** to proceed:

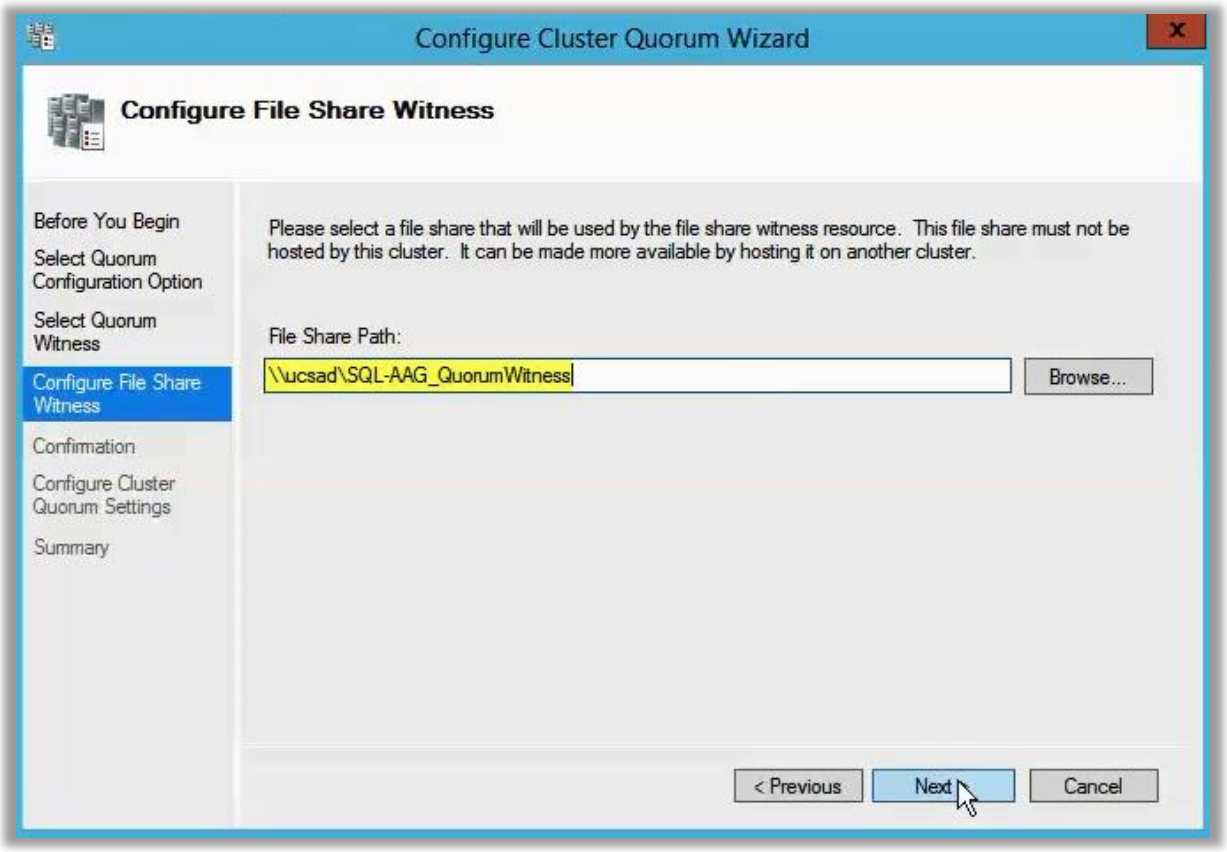


Figure 55 - Cluster Quorum Configuration Wizard - Enter the File Share path, using the FQDN of the share, if required

5. Review the confirmation page and click **Next** to continue:

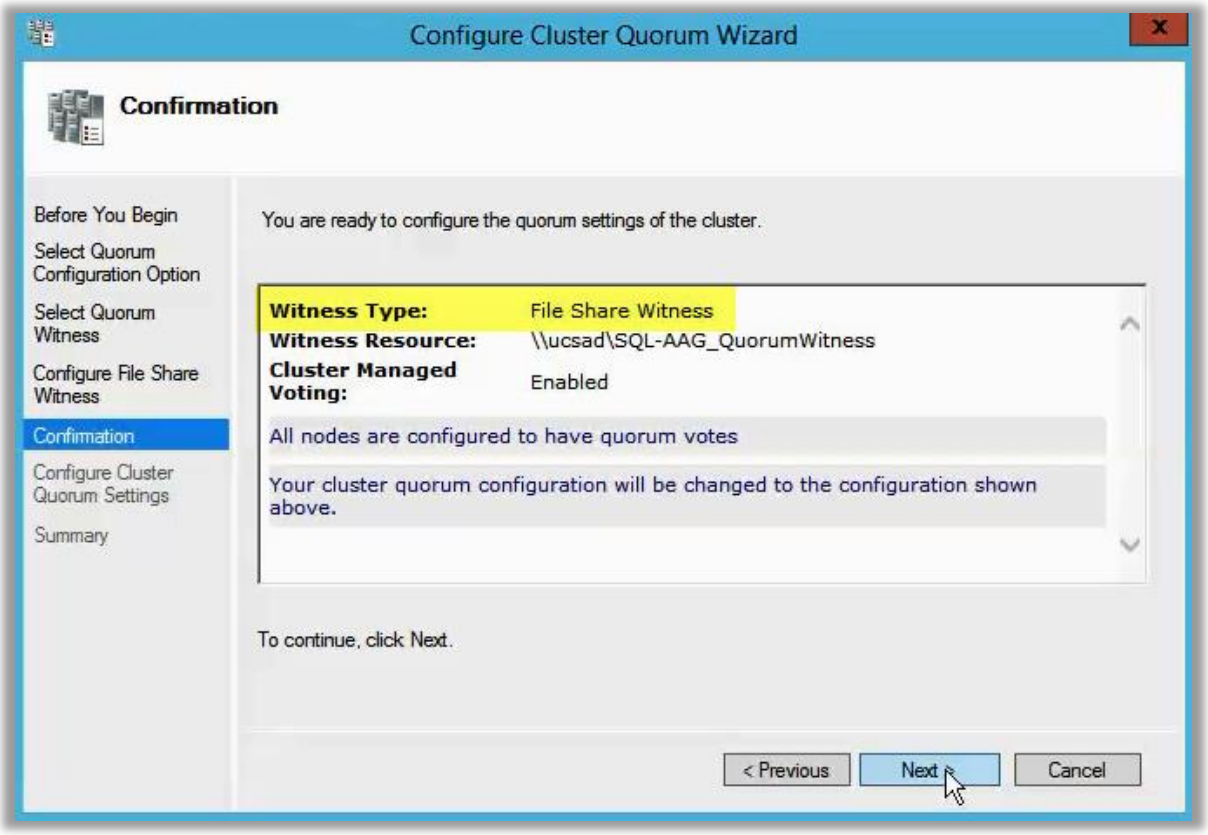


Figure 56 - Cluster Quorum Configuration Wizard - Confirmation

6. After the Quorum has been configured, click **View Report** on the summary page to review the details and ensure the operation was successful and without errors.

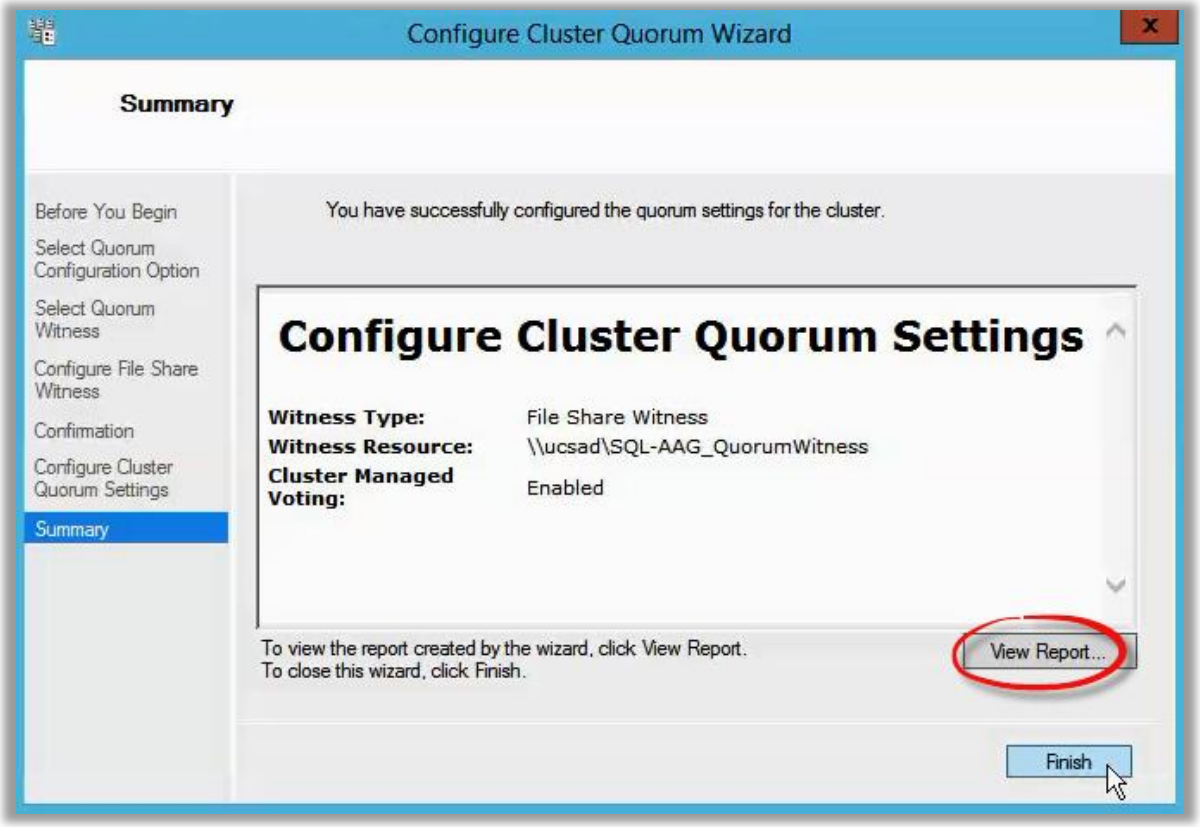


Figure 57 - Cluster Quorum Configuration Wizard - Summary

Congratulations, the Quorum is now configured!

Appendix C – Step-by-Step: Adding Databases to an existing Availability Group

Here are the steps to add a database to an existing AG. Note: this is only one of several methods available.

1. **Backup the Database** - Before adding the database to an availability group, one of the pre-requisites is that the database needs to have had a full backup. If this is not done, you will be warned in a later step and prevented from proceeding with the add operations.
   In this example, a full backup was made to a folder on the vDisk allocated for SQL Backups, and shared as \\Server\InitialSync, accessible to the other nodes. In our tests, selecting the "copy-only" option was not sufficient to meet prerequisites and a FULL backup was required. **NOTE: Make sure there is enough free space in the backup vDisk to hold the full backup (.bak) of the database(s) being added to the AG**

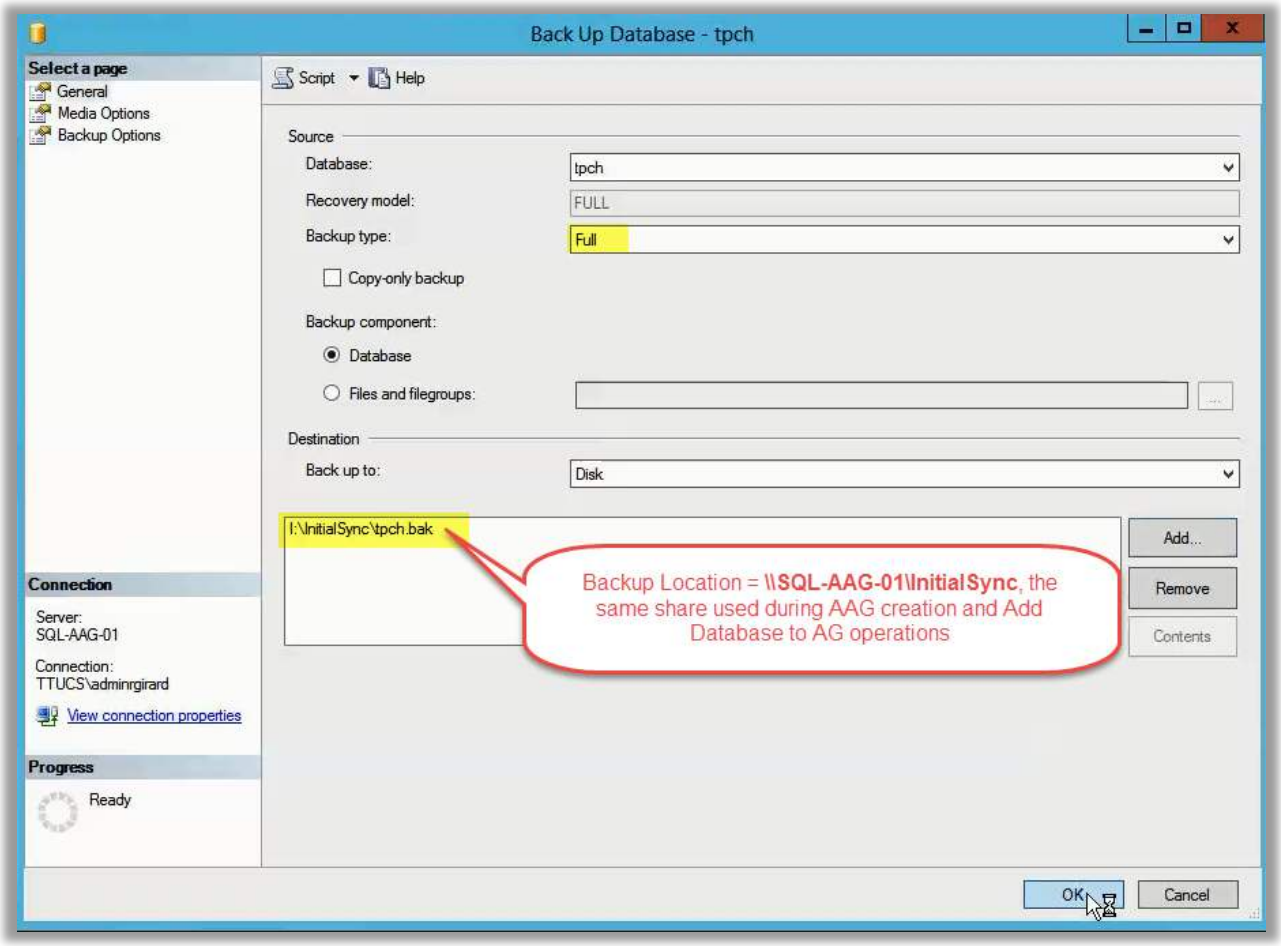


Figure 58 - Prior to adding a database to an availability group, create a full backup

2. Once the backup has completed, navigate to **AlwaysOn High Availability – Availability Groups** and **right-click** on **Availability Databases** from within SQL Management Studio. Choose **Add Database…**

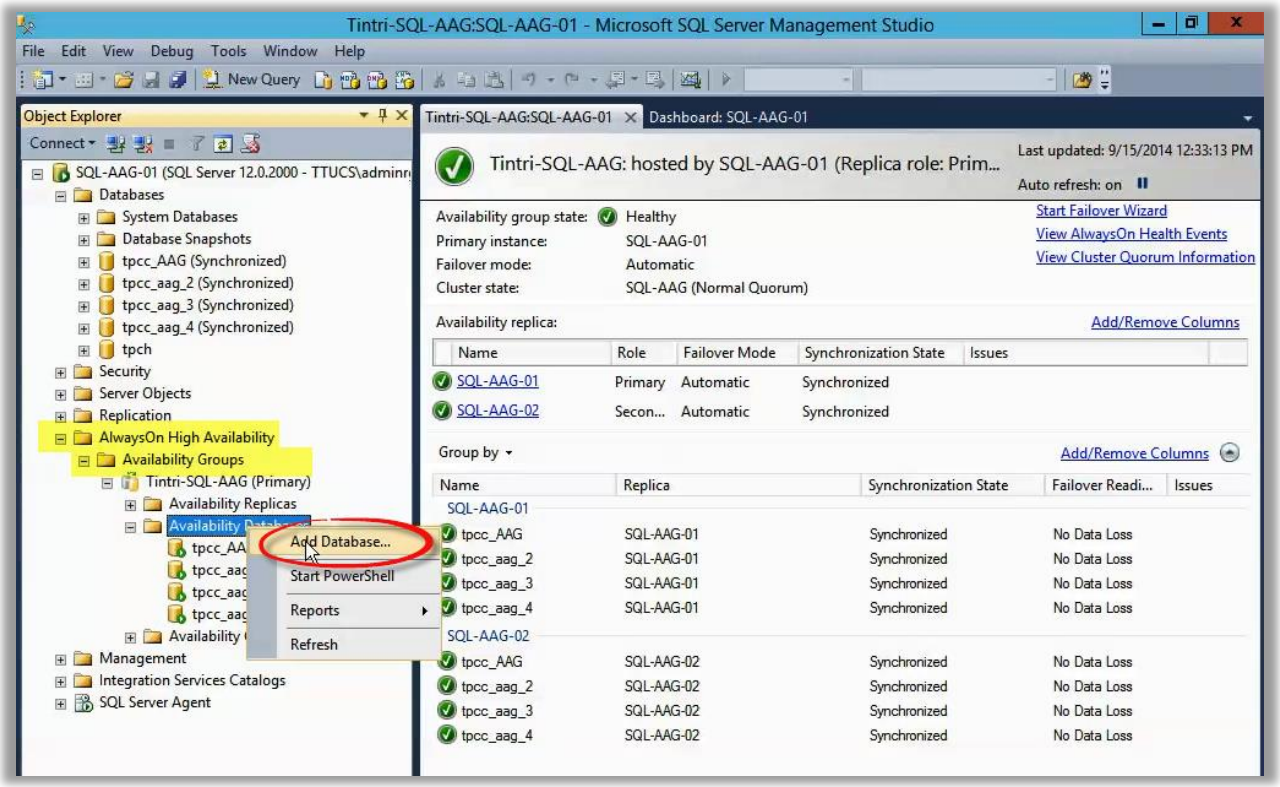


Figure 59 - The AlwaysOn Dashboard

3. The Add Database to Availability Group Wizard appears. Click **Next** to proceed.

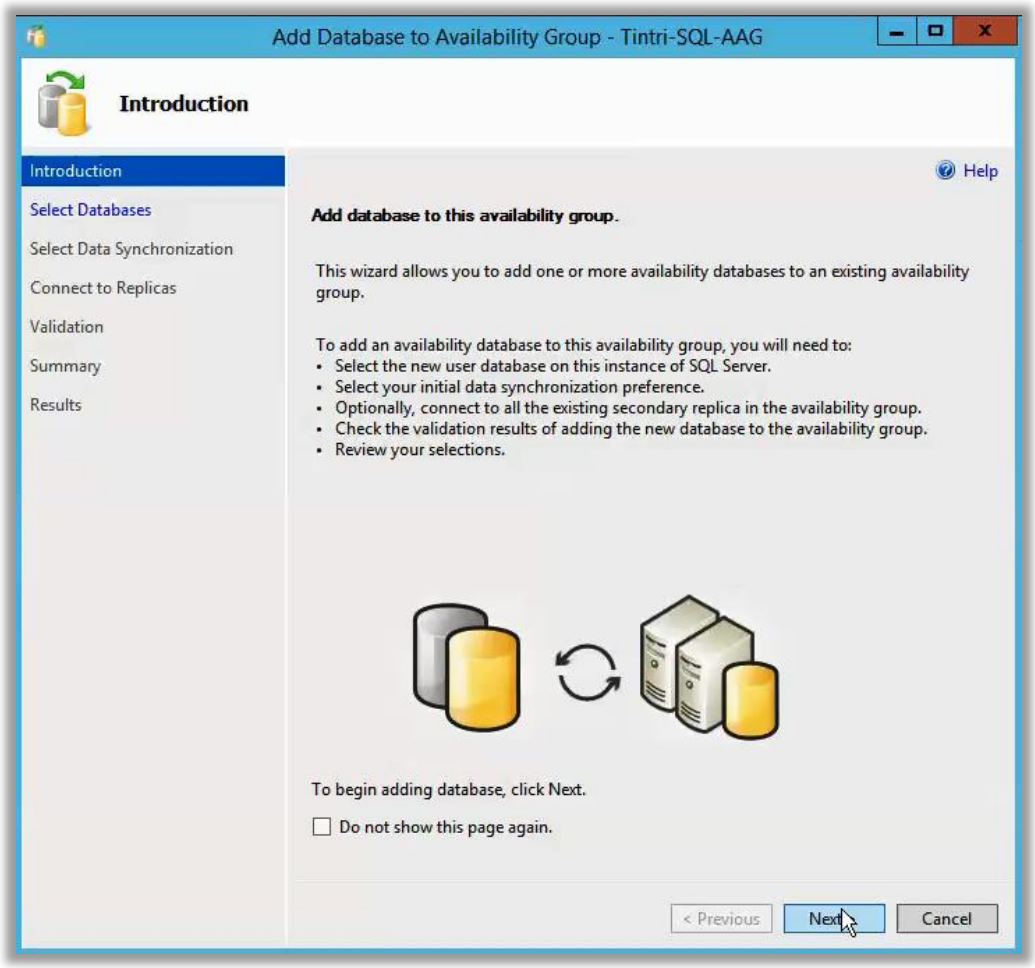


Figure 60 – Add Database to Availability Group wizard - Introduction

4. Select the database(s) you want to add. If prerequisites have not been met, click the link provided for instructions on how to proceed. In our tests, the prerequisite was always related to taking a FULL backup first. Click **Next** to proceed.

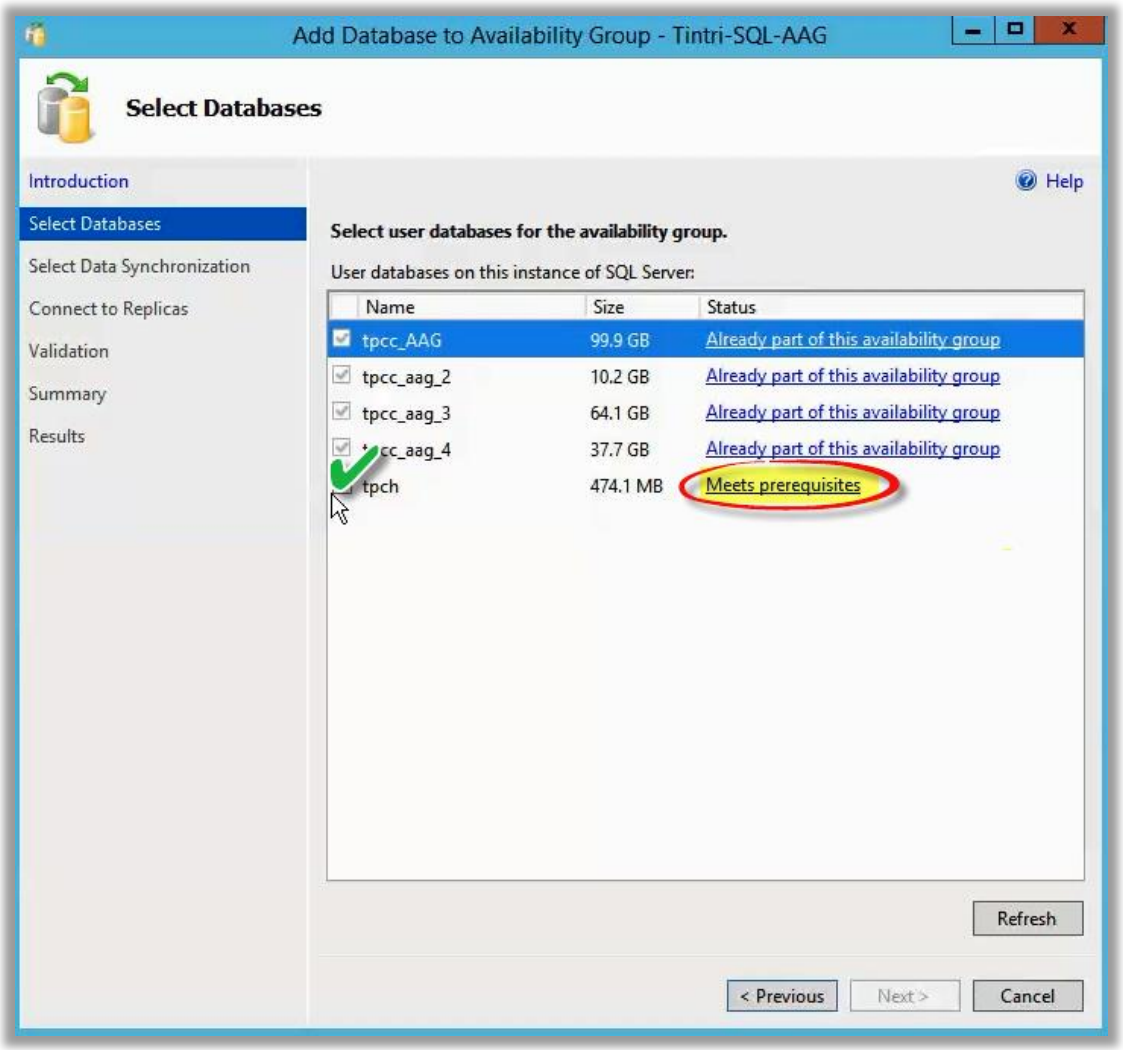


Figure 61 - Add Database to Availability Group wizard – Select Database(s)

5. Select a method populate data in the other nodes. We chose the **Full** option and used the backup share we created earlier, located on the local SQL instance: **\\SQL-AAG-01\InitialSync**. Click **Next** to continue.

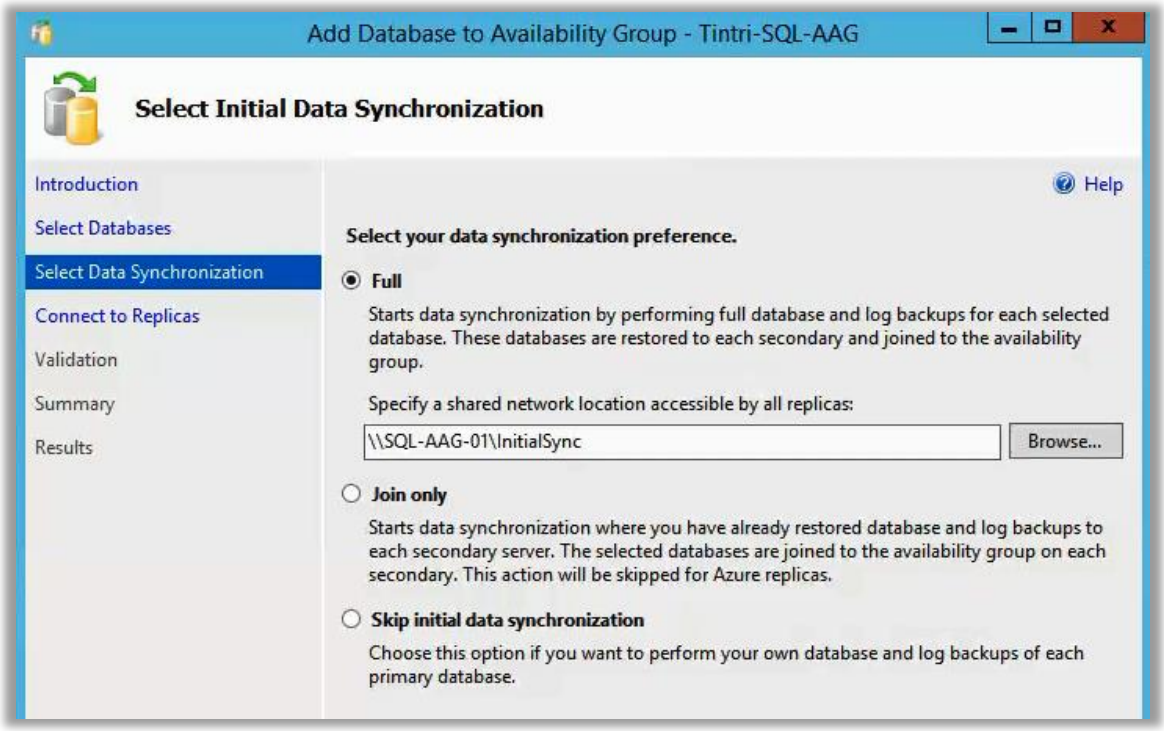


Figure 62 - Add Database to Availability Group wizard - Initial Data Synchronization options

6. **Connect to Replicas** (other nodes) within the Availability Group, and click **Next** once connected.

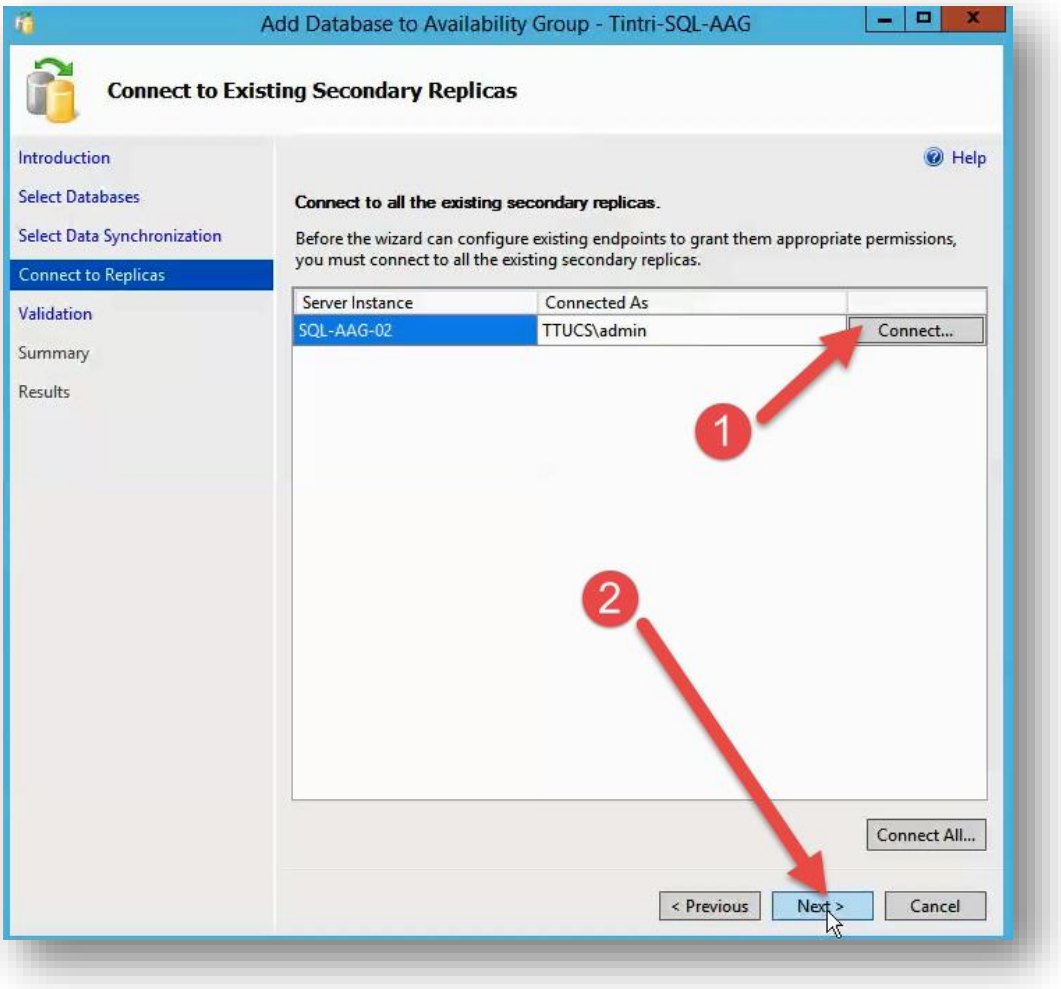


Figure 63 - Add Database to Availability Group wizard - Connect to Replicas

7. **Validation** is performed. Click **Next** assuming all tests are successful. In the event of a warning or error, review the details, correct the problem, and then continue.

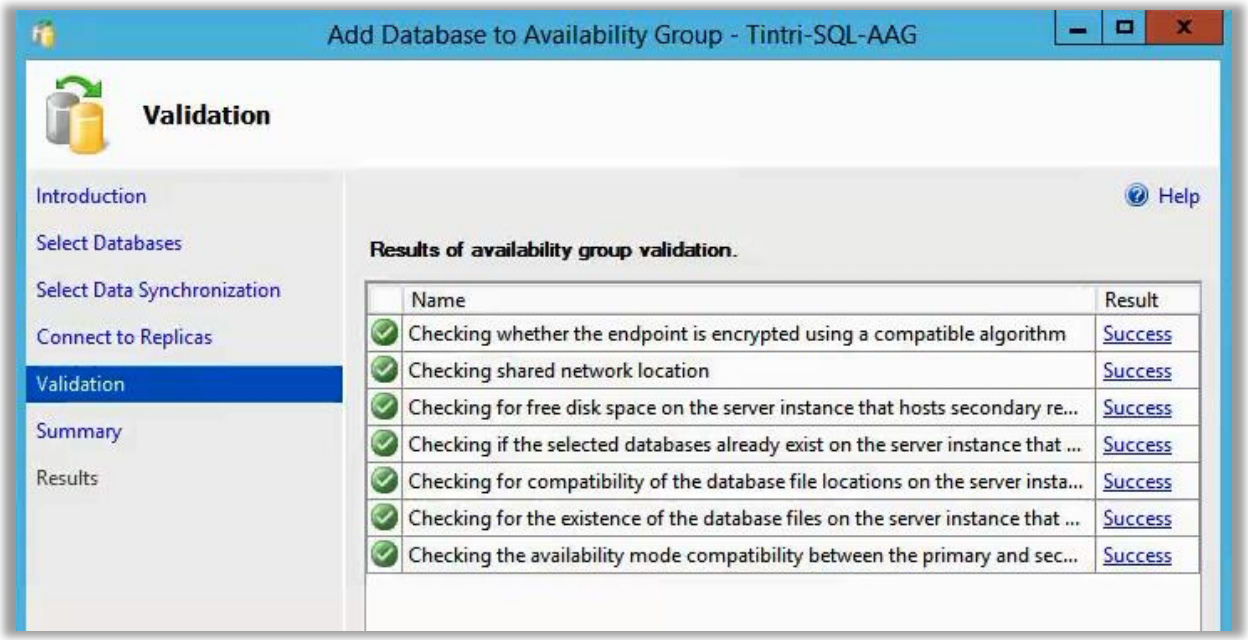


Figure 64 - Add Database to Availability Group wizard – Validation

8. **Summary** – Click **Finish** on the summary screen, assuming everything looks correct.

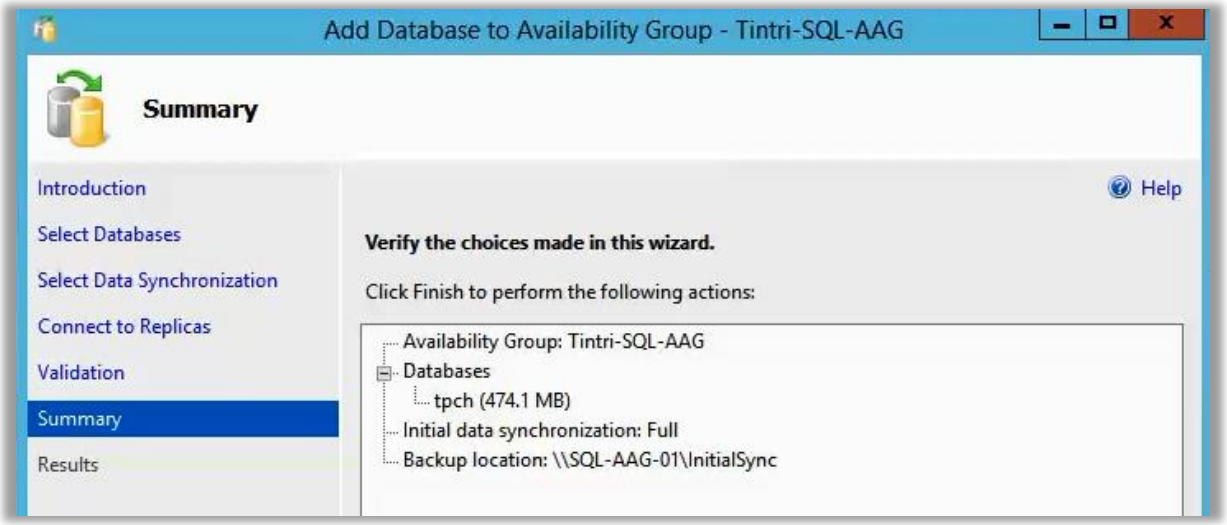


Figure 65 - Add Database to Availability Group wizard - Summary

9. As tasks are automatically performed to add the DB to the AG, click **More Detail** to track progress:

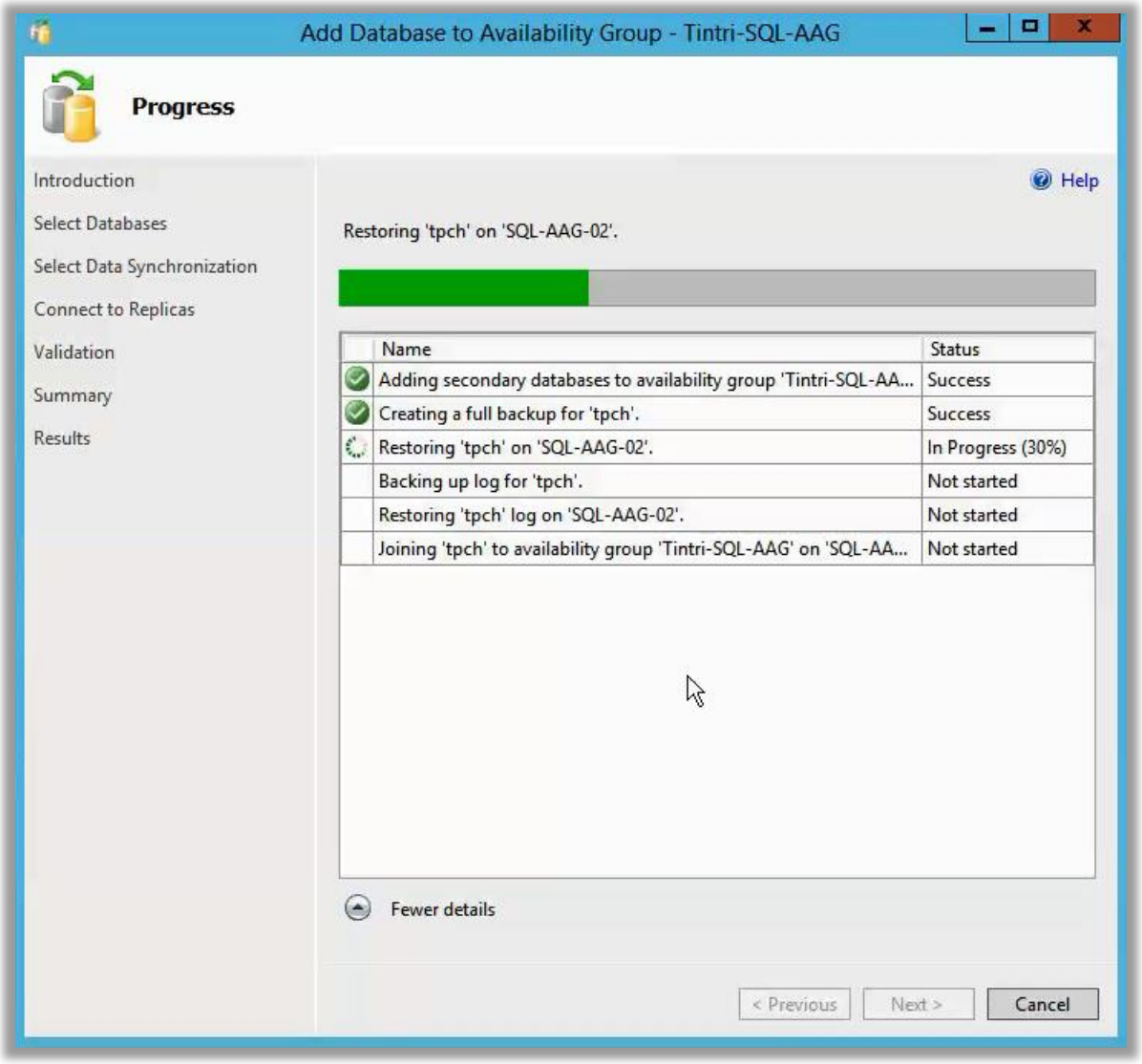


Figure 66 - Add Database to Availability Group wizard

10. In the screenshot above, the backup taken in the step before adding the DB to the availability group is restored to the other node(s). In our example, the HammerDB schema creation process is still aggressively populating the databases. All database changes from the time of the initial backup are captured and replayed into the replica server(s) and then the AlwaysOn clustering technology keeps the database(s) in sync going forward.

303 Ravendale Dr.
Mt. View CA 94043
+1 650.810.8200
info@tintri.com