Simple Talk A technical journal and community hub from Redgate

14 JANUARY 2013

Expanding AlwaysOn Availability Groups with Replication Publishers

> SQL Server 2012 AlwaysOn Availability Groups provide a high-availability and disaster-recovery solution for you SQL Server 2012 environments. Replication has been around in SQL Server for quite some time and allows you to scale out your environment. Warwick Rudd explains how to join these technologies together

Warwick Rudd 5 3

Summary

In this article we are going to be looking at the Replication Support for AlwaysOn Availability Groups that had been provided to ensure that the Replication subscribers continue to receive replicated data after an AG failover.

AlwaysOn Availability Groups provide a more granular approach to help you to meet your High Availability (HA) and Disaster Recovery (DR) requirements in increasingly complex environments.

I will not be discussing the features or installation of SQL Server 2012's AlwaysOn Availability Groups (AGs) or how to setup an environment. If you need further information into what AGs are, or how to setup your environment to use AGs, then please read my previous article – SQL Server 2012 AlwaysOn.

What is Replication ? – "Replication is a set of technologies for copying and distributing data and database objects from one database to another and then synchronizing between databases to maintain consistency". This is the definition from MSDN, if you wish to read more you can here.

Replication was first introduced as a built-in feature back in SQL Server 6.0. With the release of SQL Server 2012 some new features and improvements were added:

- Replication Support for AlwaysOn Availability Groups
- Replication Extended Events
- Support for 15,000 Partitions

Replication Support for AlwaysOn Availability Groups

SQL Server 2012 Replication supports the following new features:

- A publication database can be part of an AG. The publisher instance must share a common distributor. The types of replication that are supported within an AG are Transaction, Merge and Snapshot.
- A database in an AG secondary replica cannot be a publisher. Republishing is not supported.
- Peer-To-Peer (P2P) bi-directional, reciprocal transactional publications, and Oracle Publishing are not supported.
- A database that is enabled for Change Data Capture (CDC) can be part of an AG.
- A database enabled for Change Tracking (CT) can be part of an AG.

In order to support Replication with AGs, four new stored procedures have been provided :

Â· Sp_redirect_publisher

This new stored procedure specifies a redirected publisher for an existing publisher/database pair. If the publisher database belongs to an AG, the redirected publisher is the Availability Group Listener (AGL) name associated with the AG.

Â· Sp_get_redirected_publisher

Replication agents use this new stored procedure to query the distributor to determine whether the original publisher has been redirected. This would imply that you have had a failover of the AG on which your publisher is located.

Â· Sp_validate_redirected_publisher

This new stored procedure verifies that the current host for the publishing database is capable of supporting replication. It must be run from a distribution database.

Â· Sp_validate_replica_hosts_as_publishers

This new stored procedure is an extension of the **sp_validate_redirected_publisher**. This stored procedure validates an entire AlwaysOn replication topology. Like the stored procedure above,this one also needs to be run from a distribution database.

In order for AGs to support Replication, three replication agents were modified. The Log Reader, Snapshot and Merge Agents were modified to use the **sp_get_redirected_publisher** stored procedure to cope with determining where publisher is located.

As we are having a published database that is now supported for failover from a primary to a secondary replica, the default behaviour for the Log Reader is that it will not process any log records until those records have been hardened across all AG Replicas. This means that if we have a primary and two secondary replicas with synchronous data transfer between the primary and one secondary, as well as asynchronous data transfer between the primary and the second secondary, the log reader will not process any of the log records until all those records have been hardened on the primary and both secondaries. Because there is asynchronous data transfer between replicas, this could potentially add some latency to the hardening of records due to distance, band width, volume of traffic etc. This leads to latency of the data transfer to the subscribers.

This behaviour ensures that, in the event of a failover of the AG, the subscribers avoid getting data that is not in the new primary. There is a Trace Flag 1448 which allows us to change when the log reader reads the hardened transactions. With this flag enabled the data is allowed to be replicated to the subscriber once the transaction is hardened on the synchronous replicas. This allows for the transactions to get to the subscriber more quickly. However, in the event of a failover to an asynchronous replica, there is still a possibility that the subscriber has hardened transactions that do not exist on the publisher.

Scenario

In our environment we need to be able to provide:

- HA & DR for our application databases.
- Reporting-specific capabilities on a subset of our live transactional data.
- Specific indexes on the reporting subset of data in order to provide performance for reporting that does not impact the transactional data.

To meet these requirements and for the purpose of this article we have the following environment:

- 3 Node Windows Server Failover Cluster (WSFC)
    - Server2012Node1
    - Server2012Node2

    - Server2012Node3
- 3 Standalone default instances of SQL Server 2012
- 1 Clustered Instance of SQL Server 2012

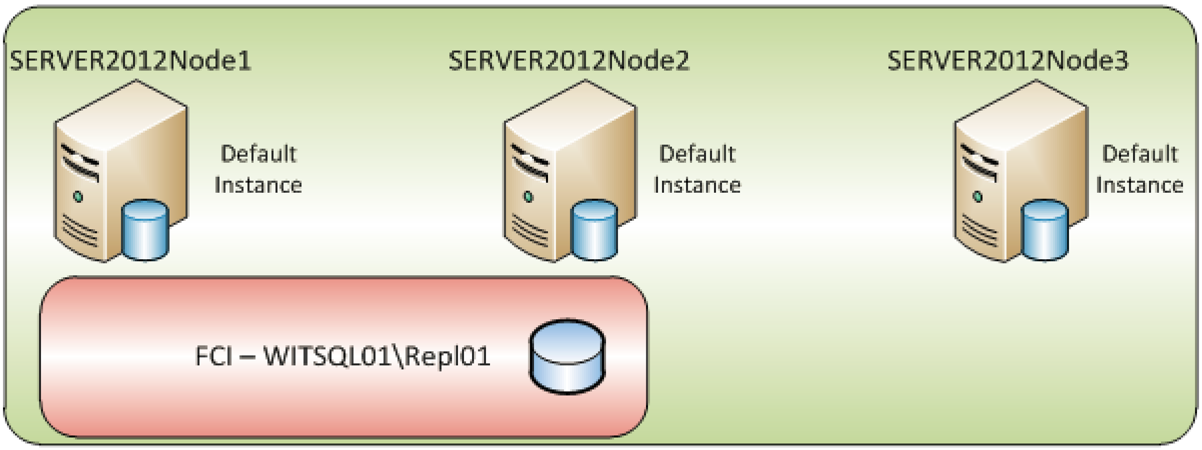


Image 1 – Environment

On **Server2012Node1** we have a database **AGTransRepl** which will be configured with Transactional Replication to a subscriber on **SERVER2012Node2**. As we are going to be adding the database into an AG we need to configure a remote distributor. To provide some HA for our remote distributor , we will place the Distributor database into the Failover Clustered Instance (FCI) **WITSQL01\Repl01**. The AG will be configured with Synchronous data transfer from **SERVER2012Node1** to **SERVER2012Node3**.

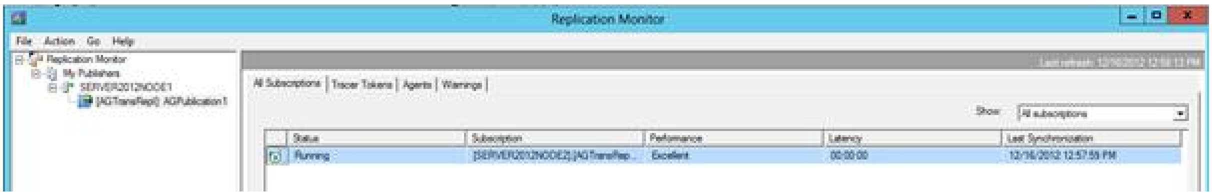
This is my development\test environment and I am limited in the number of VMs that I am able to configure and run. In a production environment, this would potentially be split into two WSFCs using a combination of traditional Failover Cluster Instances (FCI) and AlwaysOn Availability Groups (AGs). The concepts are relevant all the same.

In order to set up our environment to use transactional replication in conjunction with AGs, we need to follow these steps

1. Configure all AG Replicas as Publishers
2. Create an AG
3. Configure the publication for redirection

Confirm Transactional Replication is working

Right-Click on the Replication Tree in SSMS and then click on *Launch Replication Monitor*. Confirm the status of your newly created transactional publication.


Replication Monitor
File Action Go Help
Replication Monitor
My Publishers
SERVER2012NODE1
[AGTransRepl]: AGPublication1
Last refresh: 12/16/2012 12:58:13 PM
All Subscriptions | Tracer Tokens | Agents | Warnings
Show: All subscriptions

Status	Subscription	Performance	Latency	Last Synchronization
Running	[SERVER2012NODE2].[AGTransRep...	Excellent	00:00:00	12/16/2012 12:57:58 PM

Image 2 – Replication Monitor

Configure all AG Replicas as Publishers

Each of those replicas that are participating in an AG and will have a published database needs to be configured as a publisher. Connect to the FCI that is currently configured as a distributor. Right Click *Replication* and then click on *Select Distributor Properties*.

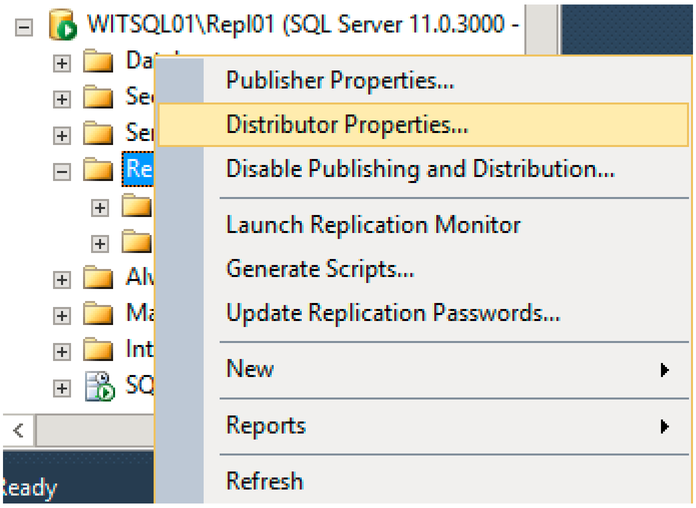
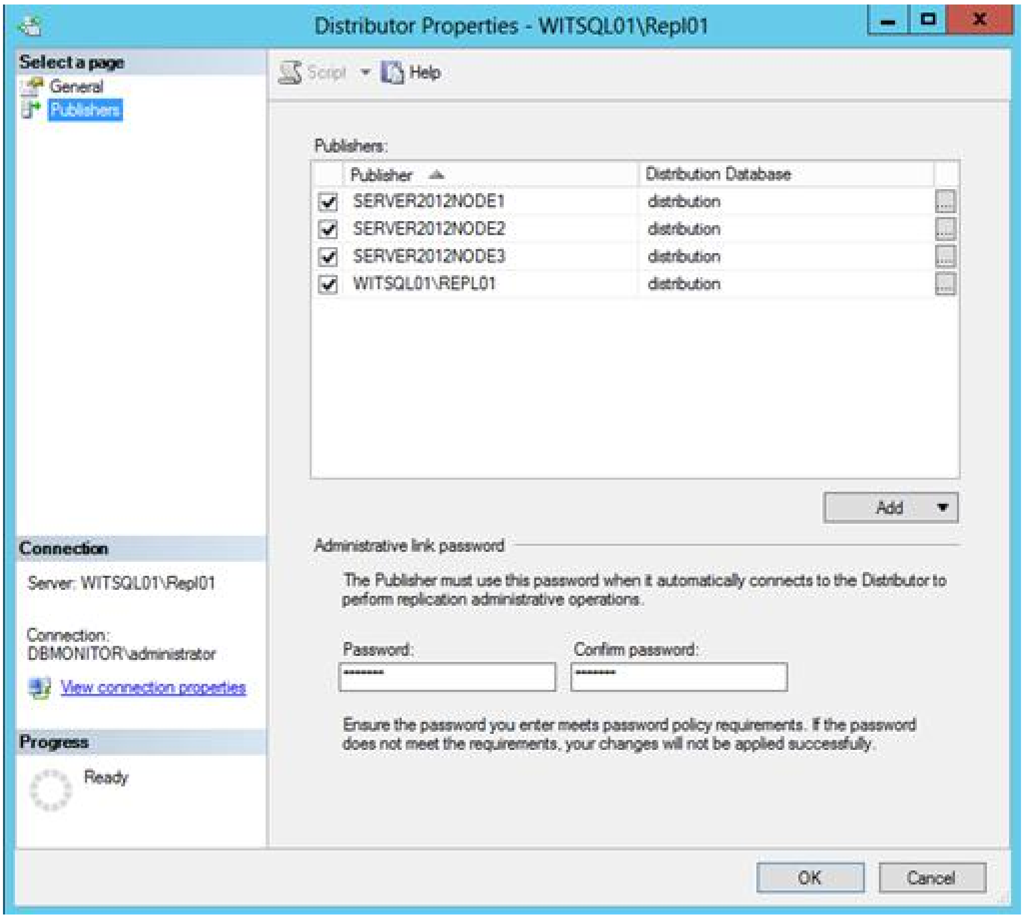

WITSQL01\Repl01 (SQL Server 11.0.3000 -
Da
Se
Se
Re
Al
Ma
Int
SQ
Ready
Publisher Properties...
Distributor Properties...
Disable Publishing and Distribution...
Launch Replication Monitor
Generate Scripts...
Update Replication Passwords...
New
Reports
Refresh

Image 3 – Distributor Properties

Click *Add* and then select *Add SQL Server Publisher*. Perform this for each replica that will be participating in the AG.

Click *Ok*.



Publishers:

	Publisher	Distribution Database
☑	SERVER2012NODE1	distribution
☑	SERVER2012NODE2	distribution
☑	SERVER2012NODE3	distribution
☑	WITSQL01\REPL01	distribution

Administrative link password

The Publisher must use this password when it automatically connects to the Distributor to perform replication administrative operations.

Password: ********

Confirm password: ********

Ensure the password you enter meets password policy requirements. If the password does not meet the requirements, your changes will not be applied successfully.

Connection

Server: WITSQL01\Repl01

Connection: DBMONITOR\administrator

View connection properties

Progress

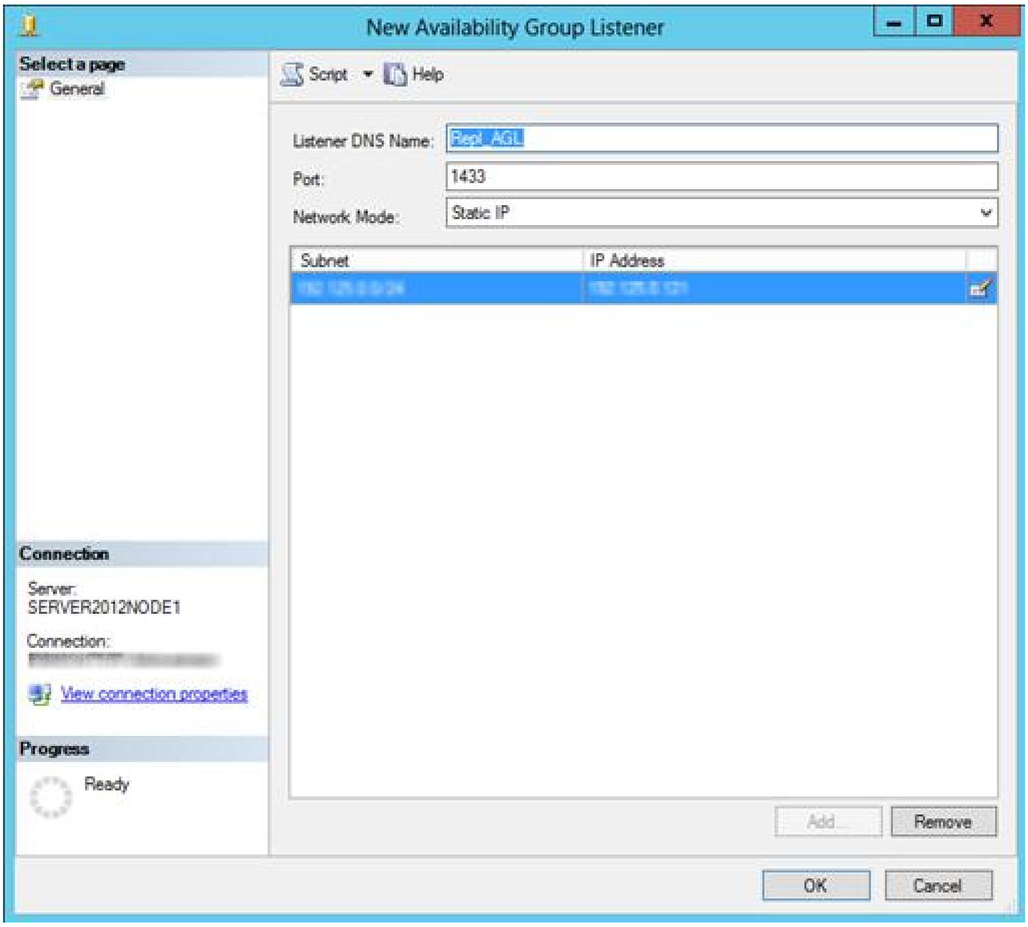
Ready

Image 4 – Distributor Replicas

Create an AG

If you need to know the steps that are required to create an availability group, you will find instructions in my previous article SQL Server 2012 AlwaysOn. When creating your AG, you have an option to create an Availability Group Listener (AGL). You will need to configure your AG with an AGL to allow you to undertake the next step in configuring your publication for redirection. If you did not configure your AGL when initially creating your AG, it can be added after the fact.

Expand AlwaysOn High Availability | Availability Groups | Expand your AG | Right Click on Availability Group Listeners | Add listener ….



Listener DNS Name: Repl_AGL

Port: 1433

Network Mode: Static IP

Subnet	IP Address
[redacted]	[redacted]

Connection

Server:
SERVER2012NODE1

Connection:
[redacted]

View connection properties

Progress

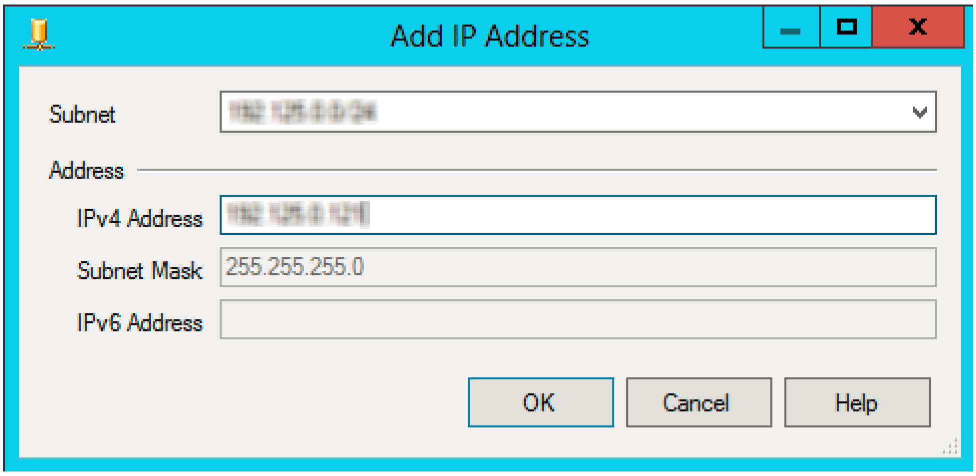
Ready

Image 5 – New Availability Group Listener

Enter your Port Number.

Click *Add*.

Enter the IP Address that will be associated with your AGL.



Subnet: [redacted]

Address

IPv4 Address: [redacted]

Subnet Mask: 255.255.255.0

IPv6 Address:

Image 6 – AGL IP Address

Click *Ok*

Now that we have created our AG with our published database in it, we need to finish off some configurations to complete our setup.

Configure the publication for redirection

The last steps we need to perform, now that our published database is residing in an AG, is to configure our replication setup for redirection. As we have already added each replica as a possible publisher, We firstly need to ensure we have set up a linked server on each replica that is pointing to the Subscription server.

```
-- Confirm that all possible publisher nodes have a linked server to the subscriber
-- If you have more than 1 possible publisher you will need to change the server
-- name in the :Connect statement below.

-- The below :Connect statement is using SQLCMD

:Connect SERVER2012Node3

Use Master
Go

Exec sp_Addlinkedserver @Server = 'SERVER2012Node2', @srvproduct = N'SQL Server';
Exec sp_serveroption @Server = 'SERVER2012Node2', @optname = N'Sub', @optvalue = N'True';

Go
```

We now need to connect to our Distribution server to configure it for redirection. This redirection allows the subscribers to reconnect to the publisher after a failover of the AG.

```
-- On the Distribution Instance of SQL Server connect and run the new system
-- stored procedure sp_redirect_publisher to associate the original publisher and
-- published database with the Availability Group Listener.
-- This tells replication that if is not able to connect to the original publisher
-- it is to connect to the Virtual Network Name (AG Listener)

-- The below :Connect statement is using SQLCMD


:Connect WITSQL01\Repl01

Use Distribution
Go

Exec sys.sp_redirect_publisher
       @Original_publisher = 'SERVER2012Node1',
       @Publisher_db = 'AGTransRepl',
       @Redirected_Publisher = 'Repl_AGL';

-- Confirm that the new meta data table has been created

Select *
From MSRedirected_Publishers;

Go

-- Validate the replication setup to confirm that all replica's in the
-- Availability Group can serve as publishers for the published db.

Declare @Redirected_publisher sysname;

Exec sp_Validate_Replica_Hosts_As_Publishers
       @Original_publisher = 'SERVER2012Node1',
       @Publisher_DB = 'AGTransRepl',
        @Redirected_Publisher = @Redirected_Publisher output;

Select @Redirected_Publisher;
```

We have now given our publisher database some high availability via SQL Server 2012s AlwaysOn Availability Groups. This means that your subscribers will continue data replication after an AG failover.

References

- Configuring Replication for AlwaysOn Availability Groups
- Supported Features with AlwaysOn Availability Groups
- SQL Server replication
- What's New (Replication)
- SP_Redirect_Publisher
- SP_Get_Redirected_Publisher
- SP_Validate_Redirected_Publisher
- SP_Validate_Replica_Hosts_As_Publishers
- Trace Flag 1448
- Replication and SQL Server Versions

Further Reading

- Replication Administration
- Maintaining an AlwaysOn Publication Database
- AlwaysOn Availability Group Interoperability
- Replication, Change Tracking, Change Data Capture, and AlwaysOn Availability Groups

Keep up to date with Simple-Talk

For more articles like this delivered fortnightly, sign up to the Simple-Talk newsletter

This post has been viewed **25972 times** – thanks for reading.

Rate

Share

Warwick Rudd

Warwick Rudd is a Microsoft Certified Master – SQL 2008, MVP, MCT and the Principal Consultant at SQL Masters Consulting (http://www.sqlmastersconsulting.com.au). Warwick is a frequent speaker at local SQL Server User Groups and SQL Saturday events in Australia and New Zealand. You can find Warwick online with his blog (http://www.sqlmastersconsulting.com.au/blog) or on Twitter (@Warwick_Rudd). When he is not playing with the SQL Server Stack, he likes to get away to the cold and spend time snowboarding.

View all articles by Warwick Rudd

Related articles

ALSO IN AVAILABILITY

Cloud Storage Replication Is Not Backup

The options that you need to select when setting up an Azure Storage service account allow you to specify the durability and high-availability of your data, but they don't provide for data recovery to a point-in-time. In fact, it means that some of the bad things that can happen to data are more efficiently replicated to all copies. Backup is quite a separate issue.… Read more

ALSO IN BI

Relational Algebra and its implications for NoSQL databases

With the rise of NoSQL databases that are exploiting aspects of SQL for querying, and are embracing full transactionality, is there a danger of the data-document model's hierarchical nature causing a fundamental conflict with relational theory? We asked our relational expert, Hugh Bin-Haad to expound a difficult area for database theorists.… Read more

ALSO IN DATABASE ADMINISTRATION

ALSO IN REPLICATION

The SQL Server 2016 Query Store: Forcing Execution Plans using the Query Store

The SQL Server 2016 Query Store can give you valuable performance insights by providing several new ways of troubleshooting queries, studying their plans, exploring their context settings, and checking their performance metrics. However, it can also directly affect the

Tribal SQL

Tribal SQL, is a reflection of how a DBA's core and long-standing responsibilities sit alongside new thinking and fresh ideas about where the DBA role is going, and what it means to be a DBA in today's businesses. … Read more

3 Comments **Simple Talk** **Login**

Recommend **Share** **Sort by Newest**

Join the discussion…

mkokoy2 • 6 months ago

Transactional with Update
Thanks. Great article.
Do you know if the transactional replication with immediate update(update by the subscriber) is supported, if the publisher will be out side of AG and the AG to be as susbscriber, but some updates are done on the subscriber. thanks again.

• Reply • Share ›

jgkiran • 2 years ago

HA with Replication
HI Warwick,

This is an exceptional article it matches with my current scenario , great work and thanks for sharing your knowledge with SQL community.

I would like to seek your advice in designing a high availability architecture , We are moving to SQL 2012 platform and we have a set of OLTP databases serving for the application usage and these databases are being replicated (transactional, every 3 min) to another box

for reporting purposes where appropriate indexing is a big requirement, I'm thinking of introducing SQL 2012 High availability groups (asynchronous) (one of the reasons to bring high availability groups in to picture is to have HA during situations like database corruption (ex: page level corruption))

Q: as the indexing on report databases is different to indexing on application dB, the failover between app db and reporting db will not provide better performance to application or reports on failover, this requirement rules out the option of using the same set of databases for failover , as you have mentioned a similar scenario in this article , can you please let me know how to handle HA in this situation ?

Indexes are my primary concern and I'm not ruling out replication completely from the design ,

I think there should be transactional replication between two sets of HA groups of oltp database and Reporting db HA Group

this helps to maintain sync after failover of oltp db, i.e I'm thinking of 4 servers two for oltp and two for reporting in HA .

Please let me know if you can suggest a better solution for this situation.

Appreciate your help.

Thanks,
Kiran

• Reply • Share ›

jgkiran • 2 years ago

HA with Replication
HI Warwick,

This is an exceptional article it matches with my current scenario , great work and thanks for sharing your knowledge with SQL community.

I would like to seek your advice in designing a high availability architecture , We are moving to SQL 2012 platform and we have a set of OLTP databases serving for the application usage and these databases are being replicated (transactional, every 3 min) to another box

for reporting purposes where appropriate indexing is a big requirement, I'm thinking of introducing SQL 2012 High availability groups (asynchronous) (one of the reasons to bring high availability groups in to picture is to have HA during situations like database corruption (ex: page level corruption))

Q: as the indexing on report databases is different to indexing on application dB, the failover between app db and reporting db will not provide better performance to application or reports on failover, this requirement rules out the option of using the same set of databases for failover , as you have mentioned a similar scenario in this article , can you please let me know how to handle HA in this situation ?

Indexes are my primary concern and I'm not ruling out replication completely from the design ,

I think there should be transactional replication between two sets of HA groups of oltp database and Reporting db HA Group

SQL

Backup and Recovery

BI

Database Administration

Database Delivery

Editor's Corner

Learn SQL Server

Performance

Reporting Services

SQL Tools

SQL Training

SSIS

T-SQL Programming

Top rated recent articles in Database Administration

The PoSh DBA: Assigning Data to Variables Via PowerShell Common Parameters

by Laerte Junior 1

SQL Server Access Control: The Basics

by Robert Sheldon 0

In-Memory OLTP – Row Structure and Indexes

by Murilo Miranda 0

SQL Server Security Audit Basics

by Feodor Georgiev 1

The SQL Server 2016 Query Store: Analyzing Query Store Performance

by Enrico van de Laar 1