SQL Server AlwaysOn High Availability

The **Always On availability** groups feature is a **high**-**availability** and disaster-recovery solution that provides an enterprise-level alternative to database mirroring. **Always On availability** groups maximizes the **availability** of a set of user databases for an enterprise.

SQL Server AlwaysOn provides a high-availability and Disaster-recovery solution for SQL Server 2012. It makes use of existing SQL Server features, particularly Failover Clustering, and provides new capabilities such as availability groups.

One of the better-known features in the release of SQL Server 2012 Enterprise Edition is AlwaysOn. This has been designed to meet the ever-increasing need for 'High Availability' (HA). AlwaysOn does not use entirely new technologies but makes more effective use of existing technologies that are tried and tested. It aims to provide more granular control to achieve High Availability. Currently, depending on your environment, you could already be using one or more of the following HA components that existed in previous versions of SQL Server:

- Single Site Windows Server Failover Clustering
- Multi-Site Windows Server Failover Clustering
- San level Block Replication
- Transaction Log Shipping
- Database Mirroring
- Transactional Replication
- Peer-to-Peer Replication

Windows Server Failover Clustering (WSFC)

Clustering technology has been around for quite some time, starting with Microsoft Clustering Services (MCS) back in NT 4.0 days.. The technology for WSFC is part of the backbone of AlwaysOn. A WSFC cluster is a group of independent servers that work together to increase the availability of applications and services. It does this by monitoring the health of the active node and failing over to a backup node, with automatic transfer of resource ownership, when problems are detected.

Although the WSFC is able to span multiple subnets, a SQL Server which is cluster-aware has not, until now, been able to support a clustered instance of SQL Server across multiple subnets: It has therefore been quite expensive to set up clustering across multiple data centres due to the WSFC requiring shared storage in both data centres as well as the block level SAN replication. This has required a lot of work with your storage vendors to get your setup correct.

Database Mirroring

Database mirroring gives you the ability to fully-synchronise databases from one instance of SQL Server to another, whether the second instance is located on the same server, a different server in the same data centre or to a server in another data centre. It can then switch roles with the mirrored database on failover. One of the problems with database mirroring is that it cannot automatically failover a group of databases that are inter-related. If you have several databases residing in an instance of SQL Server and one of those databases is failed over to the secondary location via your database mirroring setup, this database may be dependent on one or more of the other databases in the instance as

well. In this case, your application may not operate correctly. Another downside is that the mirrored database is not accessible. You can get around this by using database snapshots to give you a ‘read only’ copy.

AlwaysOn Nodes

The nodes that you will use in your SQL Server 2012 AlwaysOn solution have to be part of a WSFC. The first step we need to undertake in preparing our AlwaysOn nodes is to add the Failover Cluster Feature to each node. I’ll go into detail later on in this article.

AlwaysOn Storage

SQL Server versions prior to SQL Server 2012, being setup as clustered instance on a WSFC require the storage to be presented as shared storage. This requirement leads to the storage being more expensive and a little bit more complicated to configure and administer. With SQL Server 2012 AlwaysOn your solution does not have to utilise shared storage, but can use SAN, DAS, NAS or Local Disk depending on your budget and requirements. I suggest working with your storage providers to come up with the solution you need.

Synchronous & Asynchronous Mirroring

AlwaysOn uses SQL Server’s existing mirroring technology. Synchronous Mirroring, as its name indicates, requires the transactions to be written at both sites for the transaction to be completed. Whereas this can lead to increased latency in your system, it gives you zero data loss. AlwaysOn will support up to two secondary replicas synchronously replicated per availability group. Asynchronous Mirroring, on the other hand, is faster but increases the risk of data loss as it does not have the requirement to complete the transaction at the secondary site. One advantage that AlwaysOn has over mirroring is that it allows multiple usable secondaries of the database. Another advantage is the ability to have a combination of Synchronous & Asynchronous Mirroring in your setup so as to reach the best compromise between performance and reliability. Depending on your HA/DR requirements you possibly could have a synchronous setup to a server in your local data centre and an asynchronous setup to a server in a secondary data centre.

Availability Groups

SQL Server 2012 AlwaysOn allows for the more granular control of your environment with the introduction of AlwaysOn Availability Groups (AAG’s). AAG’s allow you to configure groups of databases that you would like to failover all together when there is a problem with the host server. When configuring your AAG’s you:

- Configure your AAG on the Primary Replica (Your AAG contains the group of DBs that you wish to group together to failover to your secondary replicas)
- You will need to configure between one and four secondary replicas, with any combination of Synchronous (Maximum of two) and Asynchronous Mirroring (Your primary replica is available for read and write connectivity, while your secondary replicas can be configured for read-only, read intent or no access)

More in-depth information on Availability Groups is covered later.

Maintenance Tasks/ Reporting

AlwaysOnallows you to use the secondary replicas that you would have created when you setup your AAGs to undertake some regular database maintenance tasks to remove some of the performance overheads from your primary production server. Some of the tasks that you could look at undertaking on a secondary replica are:

- Database Backups
    - Full Backup With Copy_Only
    - Transaction Log Backups
- DBCC CheckDB
- Reporting
- Database Snapshots