

# **Implementation of Microsoft SQL Server using ‘AlwaysOn’ for High Availability and Disaster Recovery without Shared Storage**

*Zulqarnain Hayat\*, Tariq Rahim Soomro\*\**

## **Abstract**

*It is important for any organization to handle increase in data; normally this is the job of the DBA in organization to take care of this data growth, along with its protection and its availability. High availability of data is also most important for organizations, for that purpose, shared storage is usually the solution, for example, SAN or NAS storage as Microsoft Cluster Database instances. Mostly Organization don't have shared storage in their infrastructure, which was required for Microsoft cluster SQL database instance; due to that they used other native methods to implement like log shipping or database mirroring, which was not efficient method for high-availability and disaster-recovery solution. After that, Microsoft introduced native method called SQL AlwaysOn method that could be implemented without using SAN and NAS storage method to implement high availability and disaster recovery without using shared storage. SQL AlwaysOn option is using only Microsoft cluster service and shared folder between the nodes. This paper written for DBA and System Administrators; and is implementation of Microsoft SQL Database tier step by step.*

**Keywords:** *SQL AlwaysOn, High Availability, Disaster Recovery, WSFC, Replica*

## **INTRODUCTION**

AlwaysOn is native method of Microsoft SQL Server which are giving availability of the SQL database instance and remote running SQL database instance in case of any disaster recovery (Hayat & Soomro, 2016; Nawaz & Soomro, 2013; Weiner, Mishra, & He, 2011). Using native SQL Server feature specifically failover clustering, which provided as availability group and administrative GUI interface to monitor the availability group status also. Basically, it provides business application high availability in case of any issue happened like disk failure or fault components on Database Server level. Microsoft has started from MS SQL 2012. In previous releases, mostly organizations were following native or third party solution to achieve objective as shown below (Weiner, Mishra, & He, 2011):

- Database Mirroring

---

*Correspondence:*

*\* Ankabut IT, Khalifa University of Science & Technology, United Arab Emirates. zhayatali@yahoo.com*

*\*\* College of Computer Science & Information Systems, IoBM. tariq.soomro@iobm.edu.pk*

- Log Shipping
- Multi-Site Windows Server Failover Clustering
- Peer-to-Peer Replication
- SQL Data Replication

As per authors' experience, some of above methods are very complex and not easy to implement, for example, not easy to make failback even have more software dependency if more databases running in using SQL server database instance. Microsoft SQL AlwaysOn is providing following features and capabilities (Ozar, 2010; Otey, 2011):

- Allowing backups on a secondary replica
- Creating up to 4 secondary replicas
- Employing Built-in Compression and Encryption
- Performing DBCC statements in contrast to a secondary replica
- Providing both Synchronous and Asynchronous mirroring
- Providing a logical grouping of similar databases using Availability Groups
- Using database mirroring over TCP/IP
- Using window cluster services required to make failover or failback without shared storage

### *Prior Versions of Microsoft SQL Server*

#### *Windows Clustering Service*

Microsoft clustering is an old technology, comes from Windows NT 4.0 days and known as Microsoft Clustering services; as per part of implementation of AlwaysOn, it has dependency of windows cluster service, which also called backbone of AlwaysOn of Microsoft SQL Server database 2012. Cluster group will be created on windows cluster that works as group to provide availability of applications and services. It has also quorum device, which monitor the status of cluster groups, when any node failed it move the resources to active node.

#### *Database Mirroring*

Database mirroring give us high availability with fully synchronize database from one MSSQL instance database to other instance database either in the same Server on second instance or different Server instance database; it can be also remotely deploy either same data center or different data center. One of the drawbacks is that one needs to make manually failover for each database in MSSQL instance. So it is not good option if one is having many databases for same application; for example, if some of databases have dependency to other database(s), then second mirror database is not accessible for the users, rather can provide snapshots for read only copy. There are benefits of database mirroring (Thomas, Guyer & Vance, 2017):

- Applications availability
- Automated failover (e.g. High Availability Mode)
- Easy setup and maintenance

- Instantaneous synchronization of changes
- Perform manual or automatic failover

Also in the same way, there are limitations of database mirroring (Thomas, Guyer & Vance, 2017):

- Cannot mirror a database more than once
- Limited number of databases can be mirror per SQL instance. Failover is easy, but failback is more complex based on applications connected to database instances.
- Mirrored database cannot be queried or/and backed up
- Only supports one to one Server pairings
- Synchronous mirroring is intolerant to latency and bandwidth constraints only, while asynchronous mirroring has no latency or bandwidth problems

This paper is organized as follows; section 2 material and methods will explore virtual lab environment, software requirements, hardware requirements and step by step installation and configurations; section 3 will discuss findings of this paper; and section 4 will conclude with discussions and future work.

## **MATERIAL AND METHODS**

Experimental approach has been adopted in this study. This section describe all pre and post installation methods, which includes software requirements for setting up “Virtual Lab”; followed by Hardware and software requirements in general for implementation of Microsoft AlwaysON option for HA and DR, including installation and configurations. The following are the setup of AlwaysOn option.

### *Virtual Lab Environments*

- Windows Domain Controller Setup
- Windows Hypervisor
- Windows 2012 64Bit
- Number of VMs
- Software and hardware requirements
- Network and Host Requirements

It is considered that Windows 2012 is already installed and already have windows domain controller; each node is member of this domain controller.

### *Software Requirements*

#### Software Configuration and Hardware Configuration

- Microsoft Windows 2012 64bit.
- VMware Workstation or virtual box.
- Microsoft SQL Server 2012 SP3 64bit
- 8GB RAM Each VM

- Storage = 200 GB on each VM

### Network and Host Requirements

- Public Interface Name = eth0, eth1
- Public host configuration (IPs and hostnames)
- SQLNODE1.demo.com: 192.168.3.30
- SQLNODE2.demo.com: 192.168.3.31
- Private Interfaces Name: eth2, eth3
- Private host configuration
- SQLNODE1-priv.demo.com: 192.168.2.32
- SQLNODE2-priv.demo.com: 192.168.2.33
- Cluster Name = SQLCLS
- Cluster IP(VIP) = 192.168.3.100
- Availability Group Listener = SQLDEMOLI

The following are the steps of installation process (Sack & Mishra, 2012; Mehra & Mishra, 2012):

### Enabled Windows Cluster Service on Both Database Servers

On the initial step, setting up the basic configuration of cluster services is needed, for example, adding the Windows Failover Cluster Feature and adding cluster group to all the machines running the SQL Server instances that needed to be configured as replicas. For the operating system, it is already mentioned that this study is using Windows Server 2012 R2. Following are the steps to adding windows cluster services.

- 1 Open the Server Manager Console (SMC) and select Add roles and features. This will launch the 'Add Roles and Features Wizard'
- 2 Click 'Next' until one reaches the Select Features dialog box. Select the Failover Clustering checkbox. When prompted with the Add features; that are required for Failover Clustering dialog box; click Add Features. Click 'Next' as shown in Figure 1 below:

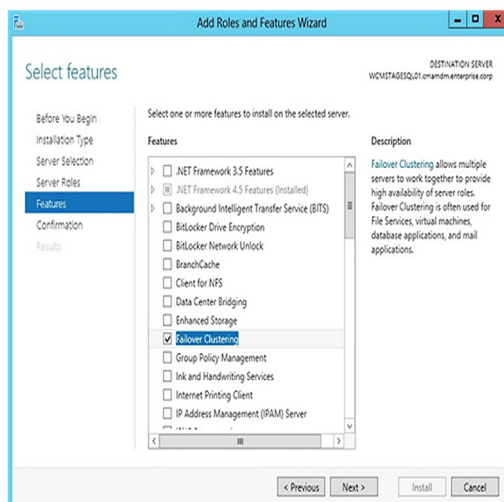


Figure 1: Add Roles and Features Wizard

- 3 Click 'Install' to install the Failover Clustering features

### *Setup Cluster Group for SQL Server Services*

As it is understood that the first task is to setup windows cluster service and user management for windows under domain user it is necessary to have both cluster nodes member of Active Directory.

- 1 Access the windows clustering application from windows Server Manager console
- 2 Validate cluster configuration right click on cluster group then validate link
- 3 Just click on next when you have opened validate wizard you will get report validation.
- 4 Add both nodes in cluster group click next
- 5 Click on Testing windows then click on next
- 6 Click confirm then next
- 7 Finish it and it will create windows cluster

Cluster validation gives several warning if shared storage like SAN or NAS are not being used but, as explained above, any shared storage is not needed, only required file share for quorum actually quorum is keeping all cluster configuration (Ray, Guyer & Vance, 2017; Sack, 2012).

- 8 To access the cluster administration console, access can be made through virtual cluster IP that is also called (VIP)
- 9 All nodes and cluster VIP hostname should be register in DNS. So all nodes or hostname should resolve. We can test using NSLOOKUP.
- 10 At the end, we get message all configuration is done successfully.
- 11 Right click on created cluster name and go to action -> more->Quorum setting
- 12 Setup the file share witness then click Next
- 13 Recommended to created one file share on domain control and shared between the cluster nodes
- 14 Then Next
- 15 Finish the wizard
- 16 Repeat all steps on second Node SQLNODE2

### *Install SQL 2012 SP2 Enterprise Edition*

Install SQL 2012 SP2 Enterprise on all SQL Servers with default configuration mentioned below:

- 1 Setup Role (select SQL Server Feature Installation)
- 2 In Feature Selection (select Database Engine Services & Management Tools Complete)
- 3 In Instance Configuration (keep defaults)
- 4 In Server Configuration (enter SQL server service account and password)
- 5 In Database Engine Configuration (Add current account and set default Database locations as shown below figure 2)

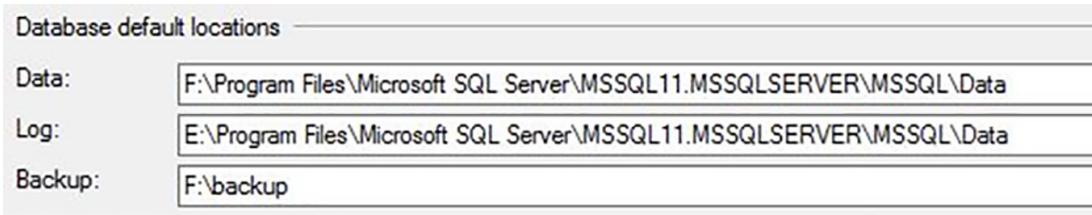


Figure 2: Database Default Location

*Setup the Always Option on Cluster Nodes*

After setting up of SQL server and cluster group, now user is ready to enable the SQL AlwaysOn option on both cluster nodes. The following are the steps:

- 1 Go to SQL configuration manager application component of SQL Server then click on SQL server property shown as Figure 3 below:

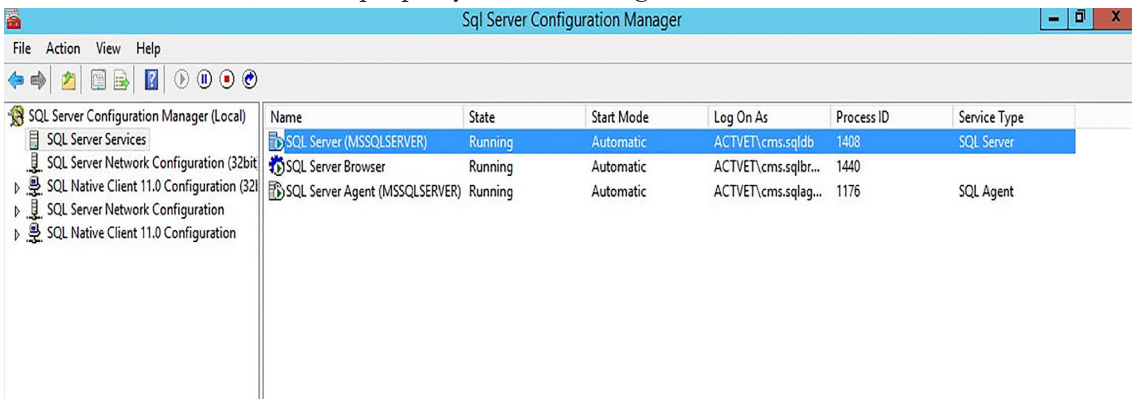


Figure 3: SQL Server Configuration Manager

- 2 Go to AlwaysOn option and check enable always option as shown in Figure 4 then click OK.

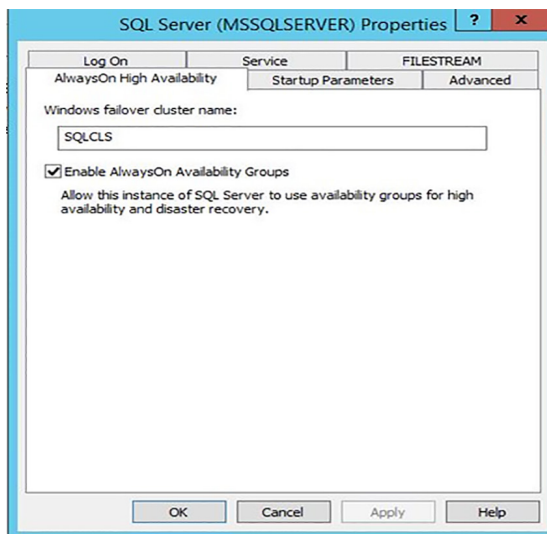


Figure 4: SQL Server (MSSQLSERVER) Properties

- 3 It will ask to reboot the SQL service

### *Setup Availability Group in Database Level*

The following are the steps need to be setup for availability group on database level. (SQLDEMOAG) following steps will be followed:

- 1 On Server “SQLNODE1” open SQL Server Management Studio. Connect to the SQL Server instance
- 2 Right click on Databases and select New Database
- 3 Enter the database name “Test”, then click OK
- 4 Expand Databases, right click on the database named “Test” then click Tasks then click Backup
- 5 Make sure the backup type is “Full” then click OK
- 6 Go to explorer then go to AlwaysOn availability group it will open the new wizard for availability group.
- 7 Click on Next
- 8 Choose the availability group for example SPAVG then click Next
- 9 Now is time to choose user databases, which need to be added in AlwaysOn availability, group (SPAVG). click Next
- 10 Expand Availability group Go to Replicas tab then click to Add Replicas button and it will ask to connect SQL server other node. Select automatic option. Go to the Endpoints tab and verify that the port number value is 5022
- 11 Go to Listener tab and create listener option and enter the following details
  - a) DNS name of listener like for example “SQLDEMOAG”
  - b) Port: 1433
  - c) Assign IP address for listener name then click OK and click Next
- 12 Next we need to setup file share where all our backup and transaction log store for setup of database availability then click Next
- 13 In this step validation, check start and gives report if getting any error during setup.
- 14 In the Summary page, verify all configuration settings and click Finish. This will create and configure the AlwaysOn Availability Group and join the databases

## **RESULTS & FINDINGS**

Microsoft SQL Server’s prior version were requiring setup as clustered instance using Microsoft Cluster Service (MCSC); and also requires the shared storage like SAN, NAS, NFS or third party in cluster group setup shared storage is cluster resource group under windows cluster name. This requirement leads to the storage being more expensive and a bit more complicated to configure and administer. With SQL Server 2012 AlwaysOn Microsoft offered solution, which does not involve shared storage, one can use SAN, DAS, NAS or Local Disk depending on one’s existing infrastructure and requirements. SQL Server 2012 AlwaysOn allows more smooth control of over environment with the introduction of AlwaysOn Availability Groups.



AlwaysOn Availability group allow to configure groups of databases in Microsoft SQL server instance, which is easy to manage and can be failover all together; which is added in Always failover group; whenever face issue with any node.

### *Discussion & Future Work*

AlwaysOn is proposed by Microsoft for SQL server 2012 database to implement High Availability and scalability and DR setup (Sack & Mishra, 2012). AlwaysOn provide protection level hardware and software it provided High availability (HA) in the same data center or remote data center, but in case of cluster, both Servers are needed in the same location. If some cluster nodes are down because of failure of software and/or hardware, even though Database here delivers HA to make sure constant data access. It could be scaled-up or down as the business scale grows. AlwaysOn is the unique setup offered by Microsoft to reduce cost and offering its capabilities and easy to setup. AlwaysOn is supposed to be used in all industries by millions of the users. If users are having mission critical application environments and who do not have shared storage to save the budgeting cost; the only solution is AlwaysOn. It is recommended that AlwaysOn Failover Cluster Instances can be used to protect data through third party shared disk solution and AlwaysOn availability group can be used to protect data through SQL Server. AlwaysOn can be extended to Azure also.

## REFERENCES

- Hayat, Z., & Soomro, T. R. (2016). Implementation of oracle real application cluster. *Review of Computer Engineering Studies, 3*(4), 81-85.
- Mehra, P., & Mishra, S. (2012). SQLCAT: SQL Server HA and DR Design Patterns, Architectures, and Best Practices Using Microsoft SQL Server 2012 AlwaysOn. Retrieved from <https://channel9.msdn.com/Events/TechEd/NorthAmerica/2012/DBI316>
- Nawaz, R., & Soomro, T. R. (2013). Role of Oracle Active Data Guard in High Availability Database Operations. *International Journal of Applied Information System, Foundation of Computer Science, New York, USA*.
- Otey, M. (2011). Microsoft SQL Server 2012: AlwaysOn Availability Groups Feature. Retrieved from <http://www.itprotoday.com/microsoft-sql-server/microsoft-sql-server-2012-alwayson-availability-groups-feature>
- Ozar, B. (2010). SQL Server 2012 AlwaysOn Availability Groups Rock - Brent Ozar Unlimited®. Retrieved from <https://www.brentozar.com/archive/2010/11/sql-server-denali-database-mirroring-rocks/>
- Ray, M., Guyer, C., & Vance, J. (2017). Database Mirroring and Log Shipping (SQL Server). Retrieved from <https://docs.microsoft.com/en-us/sql/database-engine/database-mirroring/database-mirroring-and-log-shipping-sql-server?view=sql-server-2017>



- Sack, J. (2012). AlwaysOn Architecture Guide: Building a High Availability and Disaster Recovery Solution by Using AlwaysOn Availability Groups. Retrieved from <https://www.sqlskills.com/blogs/joe/alwayson-architecture-guide-building-a-high-availability-and-disaster-recovery-solution-by-using-alwayson-availability-groups/>
- Sack, J., & Mishra, S. (2012). AlwaysOn Architecture Guide: Building a High Availability and Disaster Recovery Solution by Using AlwaysOn Availability Groups. Retrieved from <http://download.microsoft.com/download/d/2/0/d20e1c5f-72ea-4505-9f26-fef9550efd44/building%20a%20high%20availability%20and%20disaster%20recovery%20solution%20using%20alwayson%20availability%20groups.docx>
- Thomas, M., Guyer, C., & Vance, J. (2017). Always On Failover Cluster Instances (SQL Server). Retrieved from <https://docs.microsoft.com/en-us/sql/sql-server/failover-clusters/windows/always-on-failover-cluster-instances-sql-server?view=sql-server-2017>
- Weiner, M., Mishra, S., & He, M. (2011). SQL Server 2012 AlwaysOn: Multisite Failover Cluster Instance. Retrieved from [https://blogs.technet.microsoft.com/minutes\\_on\\_bi/2011/12/22/sql-server-2012-alwayson-multisite-failover-cluster-instance/](https://blogs.technet.microsoft.com/minutes_on_bi/2011/12/22/sql-server-2012-alwayson-multisite-failover-cluster-instance/)