AlwaysOn Availability Groups

Resource Links for Failover Clustering:

<https://clusteringformeremortals.com/2012/12/31/windows-server-2012-clustering-step-by-step/>

<https://clusteringformeremortals.com/2013/01/05/clustering-sql-server-2012-on-windows-server-2012-step-by-step/>

Resource Links for Availability Groups:

<https://blogs.msdn.microsoft.com/sqlalwayson/>

<https://www.brentozar.com/sql/sql-server-alwayson-availability-groups/>

<https://www.brentozar.com/archive/2012/06/why-your-sql-servers-network-connection-matters/>

<https://www.brentozar.com/archive/2013/08/common-sql-server-clustering-alwayson-and-high-availability-answers/>

<https://www.brentozar.com/archive/2012/07/planning-fail-over-sql-server-alwayson-availability-group/>

<https://www.simple-talk.com/sql/database-administration/sql-server-2012-alwayson/>

Stairway to AlwaysOn by Perry Whittle

<http://www.sqlservercentral.com/stairway/112556/>

Microsoft Docs

<https://docs.microsoft.com/en-us/sql/database-engine/availability-groups/windows/overview-of-always-on-availability-groups-sql-server>

<Https://docs.microsoft.com/en-us/sql/database-engine/availability-groups/windows/create-or-configure-an-availability-group-listener-sql-server>

<https://docs.microsoft.com/en-us/sql/database-engine/availability-groups/windows/prereqs-restrictions-recommendations-always-on-availability>

<https://docs.microsoft.com/en-us/sql/database-engine/availability-groups/windows/prereqs-restrictions-recommendations-always-on-availability#FciArLimitations>

<https://docs.microsoft.com/en-us/sql/database-engine/availability-groups/windows/failover-clustering-and-always-on-availability-groups-sql-server>

Info on Quorum Configuration:

<https://blogs.msdn.microsoft.com/sqlalwayson/2012/03/13/quorum-vote-configuration-check-in-alwayson-availability-group-wizards-andy-jing/>

<https://docs.microsoft.com/en-us/sql/sql-server/failover-clusters/windows/wsfc-quorum-modes-and-voting-configuration-sql-server#QuorumModes>

<https://www.sqlservercentral.com/Forums/Topic1533915-2799-1.aspx>

Videos:

Configure and create an AG in SQL Server 2014

<https://www.youtube.com/watch?v=VKCqRgqLAuo>

Resolve AG Listener Errors: This one addresses the permissions needed in the Computers folder of AD so that a Listener can be created via SQL Server Management Studio or T-SQL.

<https://www.youtube.com/watch?v=Z0v5nxL75SU>

Monitoring Availability Groups:

<https://docs.microsoft.com/en-us/sql/relational-databases/system-dynamic-management-views/always-on-availability-groups-dynamic-management-views-functions>

<https://docs.microsoft.com/en-us/sql/database-engine/availability-groups/windows/monitor-availability-groups-transact-sql>

<https://www.sqlskills.com/blogs/joe/answering-questions-with-the-alwayson-dashboard/>

<https://docs.microsoft.com/en-us/sql/database-engine/availability-groups/windows/use-the-always-on-dashboard-sql-server-management-studio>

<https://sqlperformance.com/2015/08/monitoring/availability-group-replica-sync>

<https://www.sqlshack.com/measuring-availability-group-synchronization-lag/>

<https://blogs.msdn.microsoft.com/saponsqlserver/2013/04/24/sql-server-2012-alwayson-part-12-performance-aspects-and-performance-monitoring-ii/>

<https://tracyboggiano.com/archive/series/monitoring-ags/>

The below query is cumulative since the last SQL Server restart so it can’t show you only what happened in the last hour, for example.

SELECT wait\_type,

waiting\_tasks\_count,

wait\_time\_ms,

wait\_time\_ms / waiting\_tasks\_count AS 'time\_per\_wait'

FROM sys.dm\_os\_wait\_stats

WHERE waiting\_tasks\_count > 0

AND wait\_type = 'HADR\_SYNC\_COMMIT';

The below articles are about detecting AG failover and related errors

<https://www.mssqltips.com/sqlservertip/3489/configure-sql-server-alerts-and-notifications-for-alwayson-availability-groups/>

<https://johnsterrett.com/tag/sql-server-2012/>

Perfmon Counters: <https://msdn.microsoft.com/en-us/library/ff878356(v=sql.120).aspx>

AG Troubleshooting and Monitoring Guide

<https://msdn.microsoft.com/en-us/library/dn135328(v=sql.110).aspx>

<https://msdn.microsoft.com/en-us/library/dn135323(v=sql.120).aspx>

<https://technet.microsoft.com/en-us/library/dn135335(v=sql.110).aspx>

<https://blogs.msdn.microsoft.com/sql_server_team/troubleshooting-high-hadr_sync_commit-wait-type-with-always-on-availability-groups/>

<https://blogs.msdn.microsoft.com/alwaysonpro/2015/01/06/troubleshooting-redo-queue-build-up-data-latency-issues-on-alwayson-readable-secondary-replicas-using-the-wait_info-extended-event/>

<https://blogs.msdn.microsoft.com/alwaysonpro/2013/12/04/recovery-on-secondary-lagging-shared-redo-target/>

Read Only Secondaries and Read Only Routing:

<https://docs.microsoft.com/en-us/sql/database-engine/availability-groups/windows/configure-read-only-access-on-an-availability-replica-sql-server>

<https://docs.microsoft.com/en-us/sql/database-engine/availability-groups/windows/configure-read-only-routing-for-an-availability-group-sql-server>

<http://www.centinosystems.com/blog/sql/availability-group-read-only-routing/>

<https://blog.rdx.com/how-to-configure-read-only-routing-for-an-availability-group/>

There is a unique URL for read only connections and a connections list that is part of the set up.  It appears that there are rules that control the read only routing based on whether a machine is a secondary or primary and this would allow for failover to a read only secondary and then a different server in the AG would server the read intent requests directed from the listener.

Monitor Performance for AlwaysOn Availability Groups

<https://msdn.microsoft.com/en-us/library/dn135338(v=sql.120).aspx>

<https://www.sqlskills.com/help/waits/hadr_sync_commit/>

Info on offloading Backups and CHECKDB:

<https://www.brentozar.com/archive/2013/12/alwayson-availability-groups-backup-checksums-and-corruption/>

* Use the WITH CHECKSUM parameter on backups - “Unless you’re doing daily DBCCs (and you’re not), then as long as you can stand the performance hit, use the WITH CHECKSUM parameter on your backups. Just doing compression alone isn’t enough.”

Each machine has its own SQL Server install.

There is no shared storage with AG.

Up to 8 secondary replicas for each availability group

Up to 3 synchronous commit replicas

Secondary replicas can be a mixture of synchronous, asynchronous and read-only.

Client apps can failover across multiple subnets almost as fast as within the same subnet.

AG Listener – The IP address used for client connections for an AG. This is synonymous with the virtual Cluster IP presented to clients when clustering is used, but with additional capabilities for routing read only traffic.

Backups can be taken from an AG secondary replica running in synchronous mode.

During the data replication of transactions if data page errors are detected those errors on the secondary are repaired.

Endpoints, the same type used in database mirroring, are used for communication between the nodes and they exist for each node of each replica.

In SQL Server 2014 the replicas stay online when the network dies.

**Transaction commit modes:**

Asynchronous commit mode on a secondary means that the replica should be in the Synchronizing state when everything is healthy. Synchronous commit mode means that the replica should be in the Synchronized state when everything is healthy. If this is not the case, then there will be warnings in the AG Dashboard.

Installing and Configuring Always On

1. From the Server manager on each server, enable failover clustering feature on all nodes involved. The feature is being enabled but failover clustering is not being set up.
2. Go to Failover Cluster Manager and create a cluster. This only has to be done one time. Each participating node does not have the “Create A Cluster” process ran on it.
3. Be sure to run the complete cluster validation test. There will be warnings concerning disks not being shared. That’s ok for an AG because no disk space is shared. The validation wizard is not NIC teaming or redundant networking aware so an error will occur there as well.

Ensure the databases that will participate in the AG are set to full recovery mode and have a recent full back up and transaction logs to be used later on the SQL instance on the other nodes in the AG.

Enabling Availability Groups in SQL Server. Before doing this step, ensure that the paths to the database files are the same on all nodes that will participate in the AG.

1. Open SQL Server Config Manager.
2. Right click the SQL Server Service account > Properties. Then on the AlwaysOn high Availability tab, select “Enable AlwaysOn Availability Groups and clock Apply, OK. This process has to be done on each SQL instance.
3. Restart the SQL Server service as required.

The steps below can be accomplished while stepping through the wizard for creating an AG. To start the wizard go to SSMS on the primary node and drill down into the node called “AlwaysOn High Availability”. Right click the folder called “Availability Groups” and choose the option for “New Availability Group Wizard”.

In the wizard you will supply the AG name, select the databases that will participate in the AG, select the failover type, whether to use synchronous commit, and whether a particular node is a readable secondary.

On the Endpoints tab of the wizard, the necessary endpoints to be created will be shown. Left click on the tab of the wizard called “Backup Preferences”. Here you will select options related to backups. To receive the full benefit of AG’s, select the option to run backups from the AG secondary. If the secondary is not available then backups will occur using the databases on the primary.

The Listener tab allows you to set up an AG Listener, but this can be done later as well. When an AG Listener is created, the Windows Virtual Network Name for the cluster is bound to the Listener name. To create a Listener later in the process, right click the AG that was created and choose “Add Listener”. Specify the chosen Listener Name and the port. Specify to use a static IP address for the Listener.

In the wizard, make a selection for one of the three methods that handle data synchronization. If you choose the “Full” option, then that share has to be accessible by all the nodes.

1. Create a mirroring endpoint on each node in the cluster and grant connect to the endpoints to the same accounts running the SQL Server service.

CREATE ENDPOINT endpoint\_mirroring

STATE = STARTYED

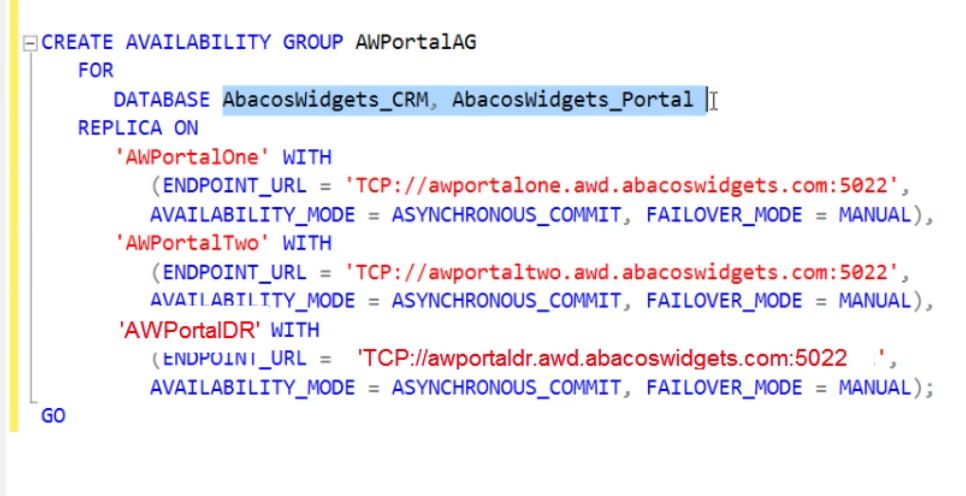
AS TCP(LISTENER\_PORT = 5022)

FOR DATABASE\_MIRRORING (ROLE = PARTNER);

GO

1. Restore a full and at least one log backup for each database in the AG to the secondary replica(s), using WITH NORECOVERY.
2. CREATE an Availability Group

This can be done with T-SQL. The below script sets things up in manual failover and asynchronous. Adjust this as needed to suit your situation.



1. On the secondary SQL instances join them to the AG.

ALTER AVAILABILITY GROUP AWPORTALAG JOIN;

1. Add the databases on the secondary instances to the availability group.

ALTER DATABASE DBName SET HADR AVAILABILITY GROUP = MYAGName

1. In the AG Dashboard on the Secondary instances, be sure to review the dashboard to ensure everything looks right.

To set up auto failover, add a listener using a static IP address and switch failover mode to automatic for the appropriate SQL Instances.

**Flexible Failover Policy**: <https://docs.microsoft.com/en-us/sql/database-engine/availability-groups/windows/flexible-automatic-failover-policy-availability-group>

Five levels – Failure\_Condition\_Level = 1,2,3,4,5

|  |  |  |
| --- | --- | --- |
| Level | Failure Condition | Transact-SQL Value |
| One | On server down. Specifies that an automatic failover is initiated when one the following occurs: | 1 |
| The SQL Server service is down. |
|  |
| [The lease of the availability group for connecting to the WSFC cluster expires because no ACK is received from the server instance. For more information, see How It Works: SQL Server Always On Lease Timeout.](http://blogs.msdn.com/b/psssql/archive/2012/09/07/how-it-works-sql-server-Always%20On-lease-timeout.aspx) |
| This is the least restrictive level. |
| Two | On server unresponsive. Specifies that an automatic failover is initiated when one of the following occurs: | 2 |
| The instance of SQL Server does not connect to cluster, and the user-specified health check timeout threshold of the availability group is exceeded. |
| The availability replica is in failed state. |
| **Three** | On critical server error. Specifies that an automatic failover is initiated on critical SQL Server internal errors, such as orphaned spinlocks, serious write-access violations, or too much dumping. | 3 |
| **This is the default level.** |
| Four | On moderate server error. Specifies that an automatic failover is initiated on moderate SQL Server internal errors, such as a persistent out-of-memory condition in the SQL Server internal resource pool. | 4 |
| Five | On any qualified failure conditions. Specifies that an automatic failover is initiated on any qualified failure conditions, including: |  |
| Detection of Scheduler deadlock. |  |
| Detection of an unsolvable deadlock. |  |
| This is the most restrictive level. |  |

**Configuring Quorom and Adding a node**

Many options here, but the one described is setting up a file share.

1. Open the Failover Cluster Manager
2. Right click the cluster name and select the option for “More Actions” > “Configure Cluster Quorum Settings.”
3. Advanced Configuration. In here you would typically take away any votes from a remote DR site because of network latency involved in such scenarios. Then click next.
4. Now you can use the option “Configure a file share witness.”

**Backing Up Availability Groups:**

Info on Backing up Secondaries

<https://msdn.microsoft.com/en-us/library/hh245119(v=sql.120).aspx>

Note: When running backups on Secondaries, the option “COPY\_ONLY” is the supported option

* BACKUP DATABASE supports only copy-only full backups of databases, files, or filegroups when it is executed on secondary replicas. Note that copy-only backups do not impact the log chain or clear the differential bitmap.
* Differential backups are not supported on secondary replicas.
* **BACKUP LOG** supports only regular log backups (the COPY\_ONLY option is not supported for log backups on secondary replicas).

A consistent log chain is ensured across log backups taken on any of the replicas (primary or secondary), irrespective of their availability mode (synchronous-commit or asynchronous-commit).

* To back up a secondary database, a secondary replica must be able to communicate with the primary replica and must be SYNCHRONIZED or SYNCHRONIZING.

Configure Backup on Availability Replicas

<https://msdn.microsoft.com/en-us/library/hh710053(v=sql.120).aspx>

<https://www.brentozar.com/archive/2015/06/how-to-configure-alwayson-ag-backups-with-ola-hallengrens-scripts/>

**Monitoring AlwaysOn**

AlwaysOn AG Dashboard in SSMS

SCOM

PerfMon

XEvents (AlwaysOn\_health)

3rd Party Tools

DMVs/Functions

**Keeping SQL Server Logins and jobs In Synch:**

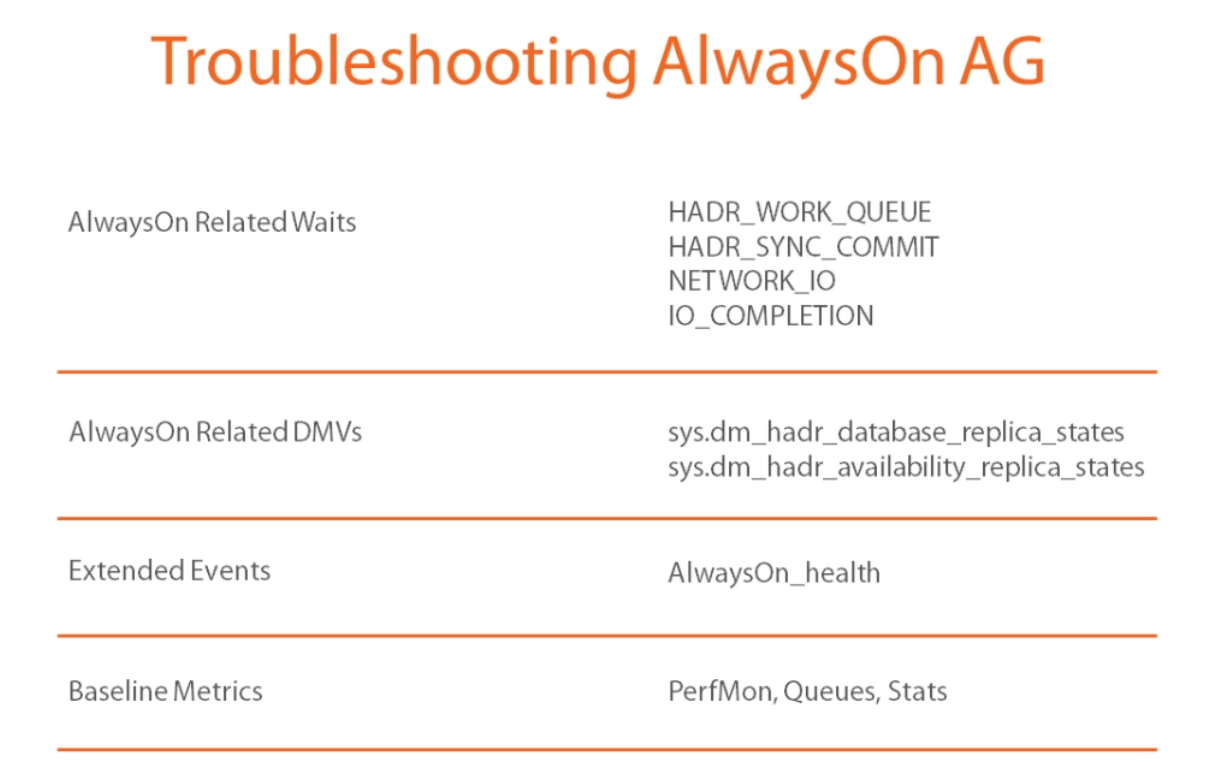
<https://docs.microsoft.com/en-us/sql/database-engine/availability-groups/windows/logins-and-jobs-for-availability-group-databases>

<http://sqlmag.com/blog/alwayson-availability-groups-and-sql-server-jobs-part-29-practical-implementation-tips>

<https://www.sqlservercentral.com/Forums/Topic1761344-3411-1.aspx>

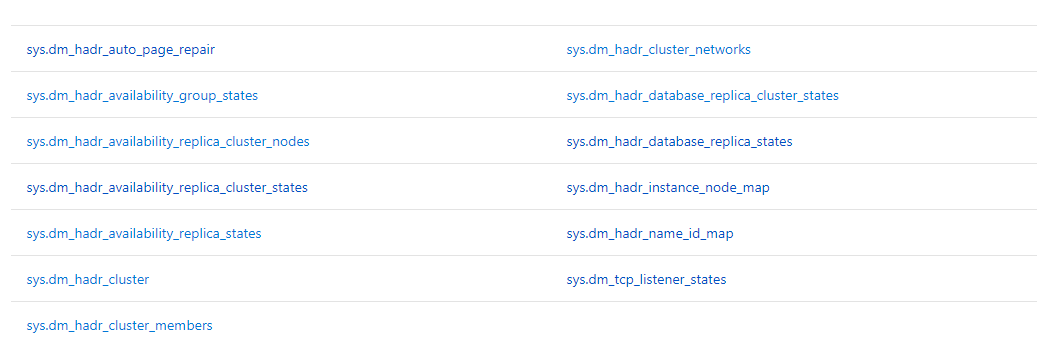
<http://www.sqldba.co.nz/blog/?p=184>

**Troubleshooting AlwaysOn AG**



AlwaysOn Related DMV’s and DMF’s and system views:

<https://docs.microsoft.com/en-us/sql/relational-databases/system-dynamic-management-views/always-on-availability-groups-dynamic-management-views-functions>



Sys.dm\_hadr\_database\_replica\_states

Sys.dm\_hadr\_availability\_replica\_states

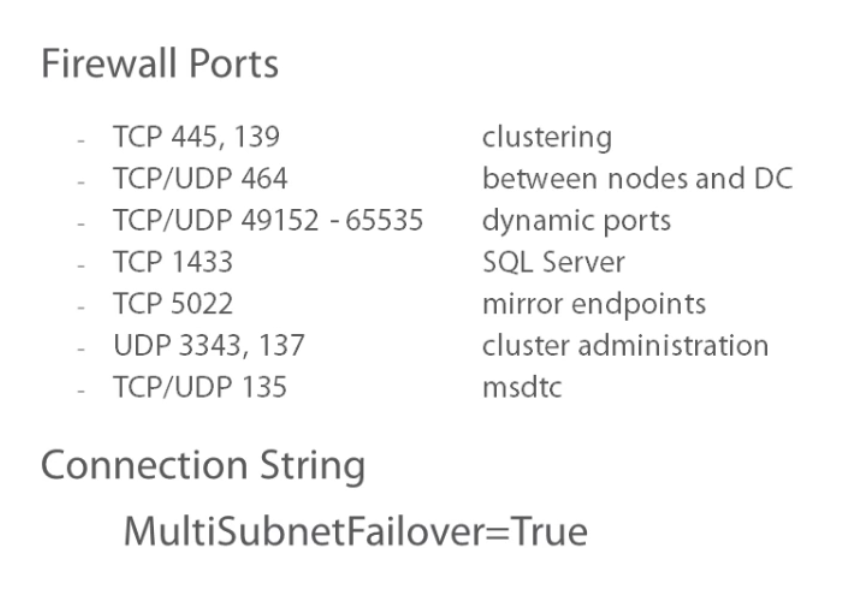
sys.fn\_hadr\_backup\_is\_preferred\_replica(@dbname)

sys.fn\_hadr\_is\_primary\_replica(@dbname)

sys.availability\_databases\_cluster

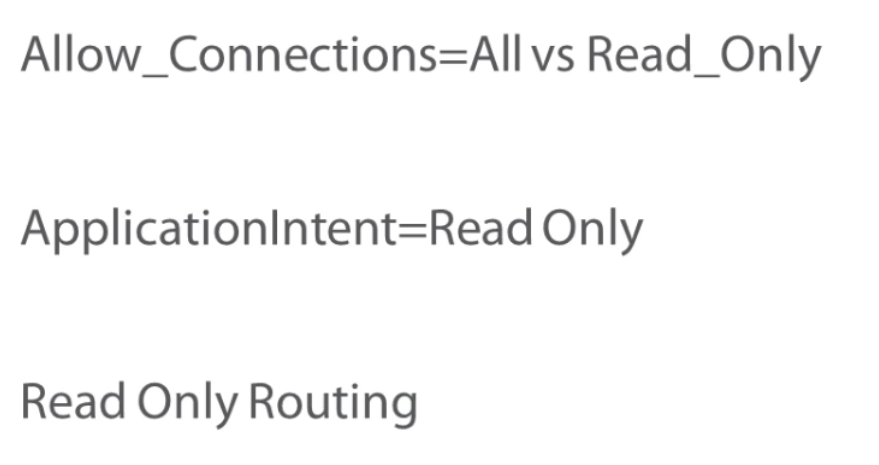
Additional AG Considerations:

Multi-subnet Failover



AG on AlwaysOn FCI – However no automatic failover available.

AlwaysOn Read Only connections: Settings and Concepts to Know



Restrictions on using the WSFC Cluster Manager: <https://docs.microsoft.com/en-us/sql/database-engine/availability-groups/windows/failover-clustering-and-always-on-availability-groups-sql-server>

Do not use the Failover Cluster Manager to manipulate availability groups, for example:

* Do not add or remove resources in the clustered service (resource group) for the availability group.
* Do not change any availability group properties, such as the possible owners and preferred owners. These properties are set automatically by the availability group.
* Do not use the Failover Cluster Manager to move availability groups to different nodes or to fail over availability groups. The Failover Cluster Manager is not aware of the synchronization status of the availability replicas, and doing so can lead to extended downtime. You must use Transact-SQL or SQL Server Management Studio.

Patching SQL Servers in an AG:

<https://blog.sqlauthority.com/2017/03/04/sql-server-apply-patch-alwayson-availability-group-configuration/>