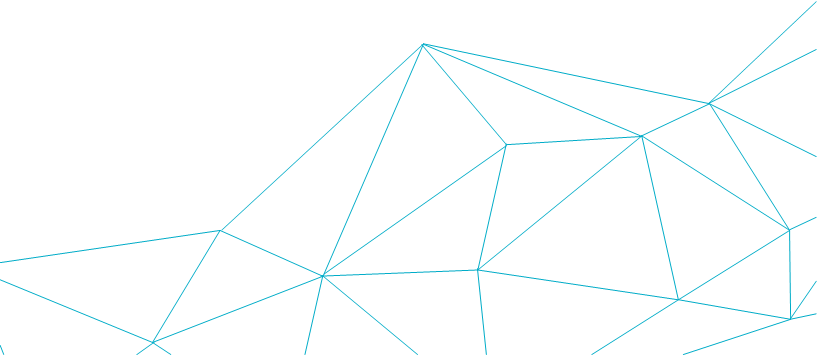


Always On Availability Groups

April 12, 2024

This document contains unpublished confidential and proprietary information of Odessa Technologies, Inc. No disclosure or use of any of these materials may be made without the express written consent of Odessa. ©1998-2020 Odessa Technologies, Inc. All Rights Reserved.



## Document revision history.

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Description | Version | Author |
| 04-10-2024 | Document Created | 1.0 | Naga Katrapati |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Introduction:

The Document has general overview of Always On Availability Group and Odessa recommendations for the databases involved in Always On Availability Group. The Document is only valid for Onpremise systems and Azure SQL VMs. For any SQL versions prior to SQL Server 2012, AOAG is not supported.

**Setup Recommendations:**

Always On Availability Groups can be configured on Windows Failover Cluster (from SQL server 2012 version). Starting SQL server 2017, Availability Groups can be setup on Standalone machines.

For High availability and Disaster Recovery, Odessa recommends setting up Always on a (minimum) 3 node Cluster.   
Below link has steps to configure AOAG.  
<https://techcommunity.microsoft.com/t5/itops-talk-blog/step-by-step-creating-a-sql-server-always-on-availability-group/ba-p/648772>

<https://learn.microsoft.com/en-us/sql/database-engine/availability-groups/windows/prereqs-restrictions-recommendations-always-on-availability?view=sql-server-ver16>

Above setup and recommendations are valid for Onpremise systems, please see below link for Azure SQL VM prerequisites and best practices.

<https://learn.microsoft.com/en-us/azure/azure-sql/virtual-machines/windows/hadr-cluster-best-practices?view=azuresql&tabs=windows2012>

<https://learn.microsoft.com/en-us/azure/azure-sql/virtual-machines/windows/availability-group-overview?view=azuresql>

Odessa supports both the traditional Availability Group and Distributed Availability Group for Disaster Recovery. If the Customer is settingup AOAG asper their standard, they can add more nodes as per their requirement.

**Setup Read Replica:**

Below link has the steps on how setup and configure Readonly routing.  
<https://learn.microsoft.com/en-us/sql/database-engine/availability-groups/windows/configure-read-only-routing-for-an-availability-group-sql-server?view=sql-server-ver16>

On a 3 node Cluster, with 2 nodes in Primary region and 1 node in another region, we recommend settingup 2 nodes in Synchronous commit with Automatic Failover mode and the third node in Asynchronous in Manual mode.

**Server Setup:**

Make sure all the nodes are setup and configured in same way. Database data files, log files, tempdb files are configured in same way on all the nodes/machines. Please make sure SQL Memory allocation, MaxDOP, Cost of threshold parallelism were set on all the nodes as per the Odessa recommendation.

**Listener Setup:**

Create listener for the Availability Group as per the Microsoft recommendations (Onpremise servers)

<https://learn.microsoft.com/en-us/sql/database-engine/availability-groups/windows/availability-group-listener-overview?view=sql-server-ver16>

For Azure SQL VM, create a DNN/VNN listener using the link below

<https://learn.microsoft.com/en-us/azure/azure-sql/virtual-machines/windows/availability-group-load-balancer-portal-configure?view=azuresql>

**Change Fillfactor:**

Using Read replica in AOAG will add overhead to tempdb and add row versioning on Primary Replica. Please make sure fill factor for all the indexes have been changed. If not, Execute Changefillfactor SP on Primary database. Make sure Tempdb is sized appropriately and files are stored in a faster disk. Row versioning will add additional overhead on tempdb.

Note: Please make sure this Changefillfactor is executed only during the maintenance window or when downtime is allowed. The SP could make database size grow, it is recommended to run this in Test environment to get to know the final size of the database.

<https://learn.microsoft.com/en-us/sql/database-engine/availability-groups/windows/active-secondaries-readable-secondary-replicas-always-on-availability-groups?view=sql-server-ver16>

**Data Latency:**

Note that Synch is near real time and there could be a latency in milli seconds to seconds, depending on Infrastructure and network. There could be high latency if there’s a bottleneck on Secondary Replica servers or if the Redo thread is blocked.

Microsoft recommends not to run any time sensitive reports on Read Replica database. below query can be used to get the latency between Primary and Readonly.

Select DB\_NAME(database\_id) as DBNAME,

datediff(ms,last\_redone\_time,last\_hardened\_time) as [MilliSeconds behind]

,Cast((datediff(mi,last\_redone\_time,last\_hardened\_time)) as decimal(18,2)) as [Minutes behind]

,redo\_queue\_size as [Redo Queue size in KB]

,redo\_rate as [Redo Rate KB/second]

, getdate() as [datetime]

From sys.dm\_hadr\_database\_replica\_states

Where 1=1 and is\_primary\_replica = 0

**Enable Querystore:**

Make sure Querystore is enabled for Odessacore database (in Primary Replica) to troubleshoot any issues. Starting SQL server 2022, Querystore is supported on Read Replica aswell. If the Database is in SQL 2022, enable Querystore on both Primary and Read Replica databases.

**SQL Agent Jobs:**

Create SQL agent jobs on all the Nodes/Replicas. On Primary Replica, jobs will be active. All the jobs on Secondary Replica needs to be disabled state. They can be enabled manually after failover.

Alternatively add the script below to first step of all the jobs and enable them. Jobs on all the nodes will run as scheduled, they will only be successful on Primary Replica. On Secondary Replica servers, all the jobs will exit/fail.

DECLARE @ServerName NVARCHAR(256) = @@SERVERNAME

DECLARE @RoleDesc NVARCHAR(60)

SELECT @RoleDesc = a.role\_desc

FROM sys.dm\_hadr\_availability\_replica\_states AS a

JOIN sys.availability\_replicas AS b

ON b.replica\_id = a.replica\_id

WHERE b.replica\_server\_name = @ServerName

IF @RoleDesc = 'Secondary'

BEGIN

RAISERROR ('Current node is Secondary Replica', 16, 1)

END

**Change Data Capture:**

Change data capture is supported with AOAG. On OdessaCore database, to enable please follow the steps from the installation documentation.

To disable CDC at database level on versions prior to SQL Server 2016 SP2, please follow below steps.

<https://learn.microsoft.com/en-us/sql/database-engine/availability-groups/windows/replicate-track-change-data-capture-always-on-availability?view=sql-server-ver16>

**Login Management:**

Make sure logins are created on all the Replicas. User mappings and permissions to user database will get automatically synched post failover. Use the below link to create logins on Secondary Replica servers.

<https://learn.microsoft.com/en-us/troubleshoot/sql/database-engine/security/transfer-logins-passwords-between-instances>

**Setup Connection string:**

For Primary Replica connection string, it is recommended to add “MultiSubnetFailover = True”. For Read Replica, add “MultiSubnetFailover = True” and “ApplicationIntent= Readonly”. Use Listener name and port number instead of server names in the connection string.

Make the following changes in the connection strings for Primary and Read Replica:

**Replica:** <add name="LeasewaveRepositoryReplica" connectionString="data source=ListenerName,Portnumber; Initial Catalog=DatabaseName;User ID=LoginName;Password=EnterPassword;Max Pool Size=300;MultipleActiveResultSets=true;ApplicationIntent=ReadOnly;MultiSubnetFailover=True;Enlist=False" providerName="System.Data.SqlClient" />

**Primary:** <add name="LeasewaveRepository" connectionString="data source= ListenerName,Portnumber; Initial Catalog=DatabaseName;UserID=LoginName;Password=EnterPassword;MultiSubnetFailover=True;  
MultipleActiveResultSets=true" providerName="System.Data.SqlClient" />