

# Troubleshooting SQL Server Always On Availability Groups

By: [Ahmad Yaseen](#)

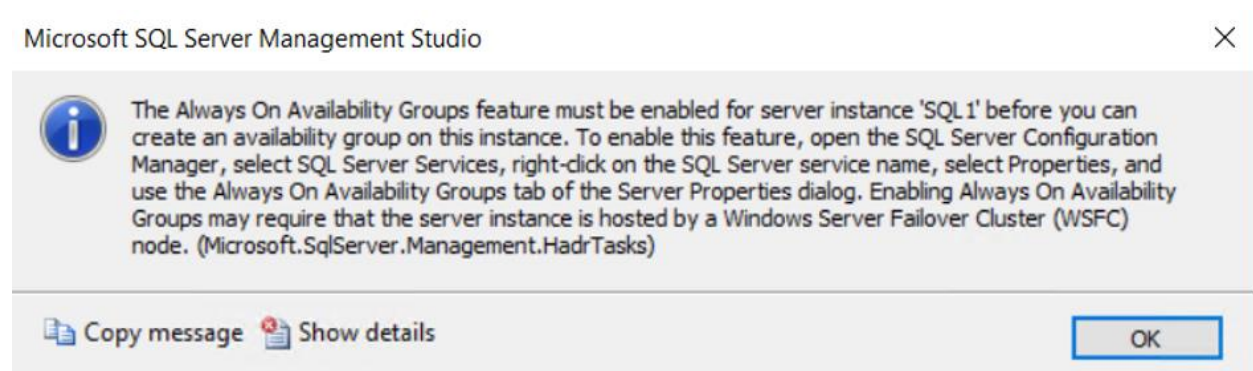
In this article, we will discuss several issues that you may face while creating, configuring or maintaining an Always on Availability Group site.

Before going through this article, it is recommended to read the previous article, [Setting up and Configuring Always on Availability Group in SQL Server](#), to be familiar with the Always on Availability Group concept and the New Availability Group wizards shown in this article.

## Always on Availability Group Feature Not Enabled

Assume that, while trying to create a new Always on Availability Group, from the Always On High Availability node, under the Object Explorer of the SQL Server Management Studio, you faced the error message below:

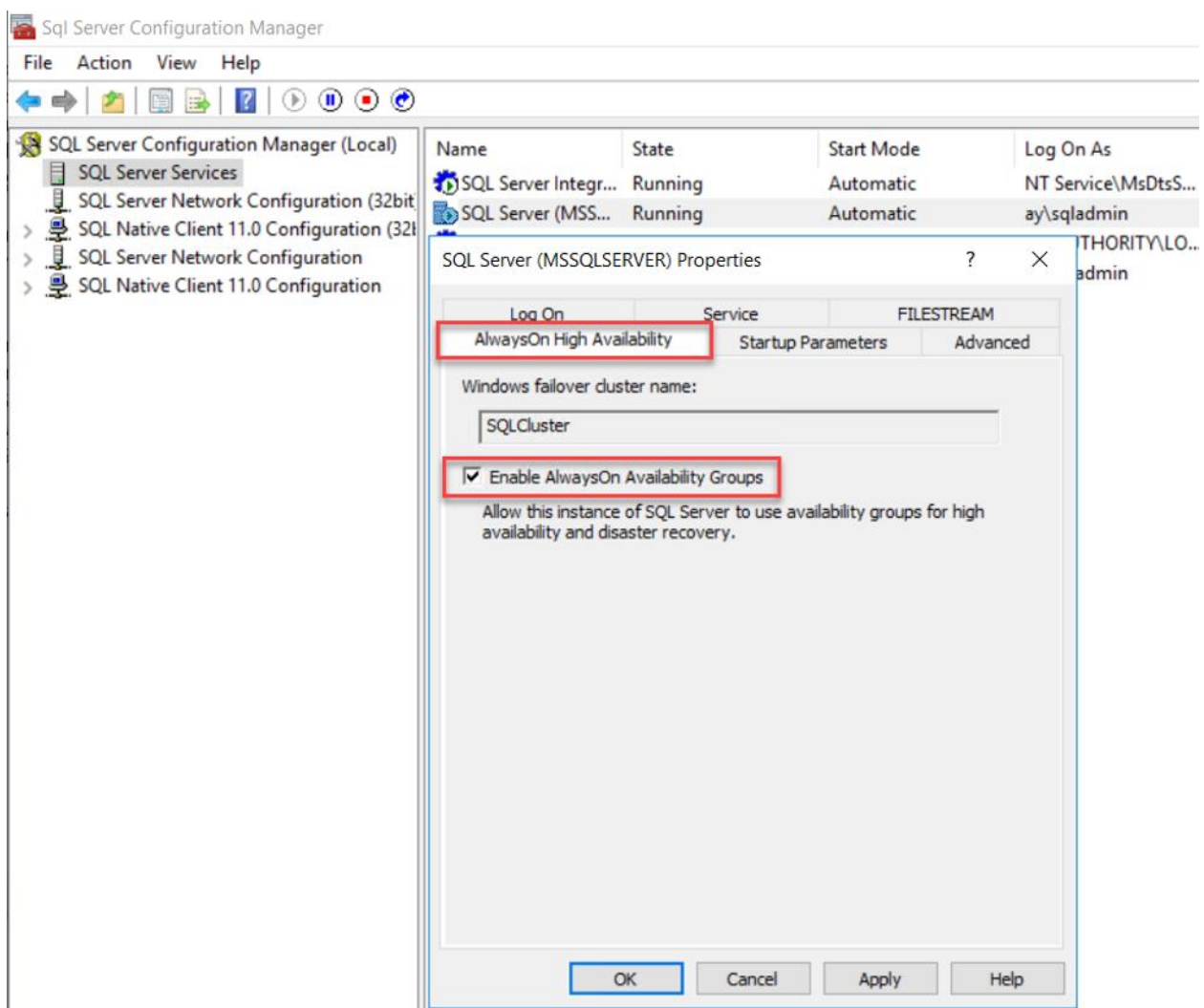
*The Always On Availability Groups feature must be enabled for server instance 'SQL1' before you can create an availability group on this instance. To enable this feature, open the SQL Server Configuration Manager, select SQL Server Services, right-click on the SQL Server service name, select Properties, and use the Always On Availability Groups tab of the Server Properties dialog. Enabling Always On Availability Groups may require that the server instance is hosted by a Windows Server Failover Cluster (WSFC) node. (Microsoft.SqlServer.Management.HadrTasks)*



It is clear from the error message that, the AlwaysOn Availability Groups feature should be enabled on each SQL Server instances that participate in the Always on Availability Group site, before creating that site.

You can easily enable the Always on Availability Group feature, by opening the SQL Server Configuration Manager console, browse the SQL Server Services tab then right-click on the SQL Server Database Engine service and choose the Properties option.

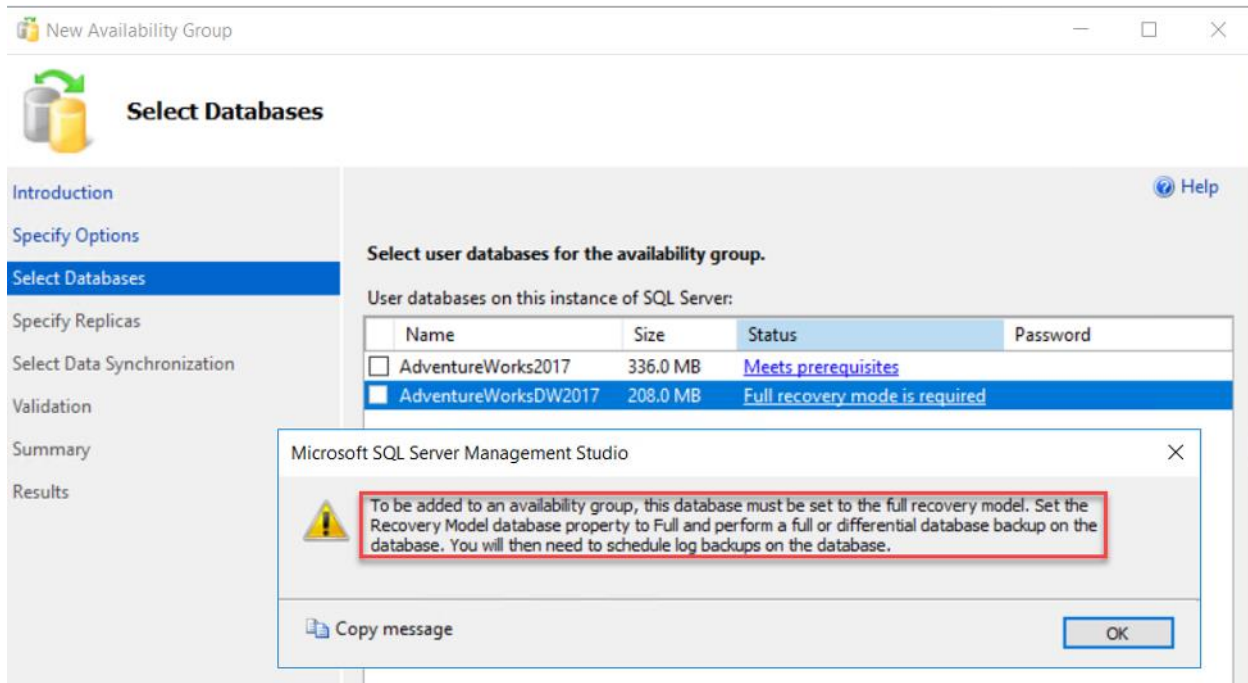
From the opened SQL Server Properties window, move to the Always on High Availability tab and check the checkbox beside the **Enable Always on Availability Group**, taking into consideration that this change requires restarting the SQL Server service to take effect, as shown below:



## Database Pre-requisites Validation Issue

In the earlier steps of the New Availability Group wizard, you will be asked to specify the database(s) that will participate in the Always on Availability Group. Before adding the database, the database should pass the pre-requisites validation check. Otherwise, the database cannot be selected from the databases lists, as shown in the error message below:

*To be added to an availability group, this database must be set to the full recovery model. Set the Recovery Model database property to Full and perform a full or differential database backup on the database. You will then need to schedule log backups on the database.*



The message is clear. Where the database should be configured with a Full recovery model and a Full or Differential backup should be performed on that database.

Also, the wizard warns you to schedule a transaction log backup for that database after changing the recovery model to Full, to truncate the transaction log file automatically and prevent running that transaction log file out of free space.

To fix that issue, change the database recovery model from Simple to Full, from the Options tab of the database properties window, then take a Full backup from that database, as shown below:

Database Properties - AdventureWorksDW2017

Select a page

- General
- Files
- Filegroups
- Options
- Change Tracking
- Permissions
- Extended Properties
- Mirroring
- Transaction Log Shipping
- Query Store

Script Help

Collation: SQL\_Latin1\_General\_CP1\_CI\_AS

Recovery model: Full

Compatibility level: Full

Containment type: Bulk-logged

Other options: Simple

Database Scoped Configurations

Legacy Cardinality Estimation	OFF
Legacy Cardinality Estimation For Secondary	PRIMARY
Max DOP	0
Max DOP For Secondary	
Parameter Sniffing	ON
Parameter Sniffing For Secondary	PRIMARY
Query Optimizer Fixes	OFF
Query Optimizer Fixes For Secondary	PRIMARY

FILESTREAM

FILESTREAM Directory Name	
FILESTREAM Non-Transacted Access	Off

Misc

Allow Scripting	True
Hide File Settings	False

Miscellaneous

Allow Snapshot Isolation	False
ANSI NULL Default	False

Allow Snapshot Isolation

Ready

OK Cancel

Back Up Database - AdventureWorksDW2017

Select a page

- General
- Media Options
- Backup Options

Script Help

Source

Database: AdventureWorksDW2017

Recovery model: FULL

Backup type: Full

☐ Copy-only backup

Backup component:

☒ Database

☐ Files and filegroups:

Destination

Back up to: Disk

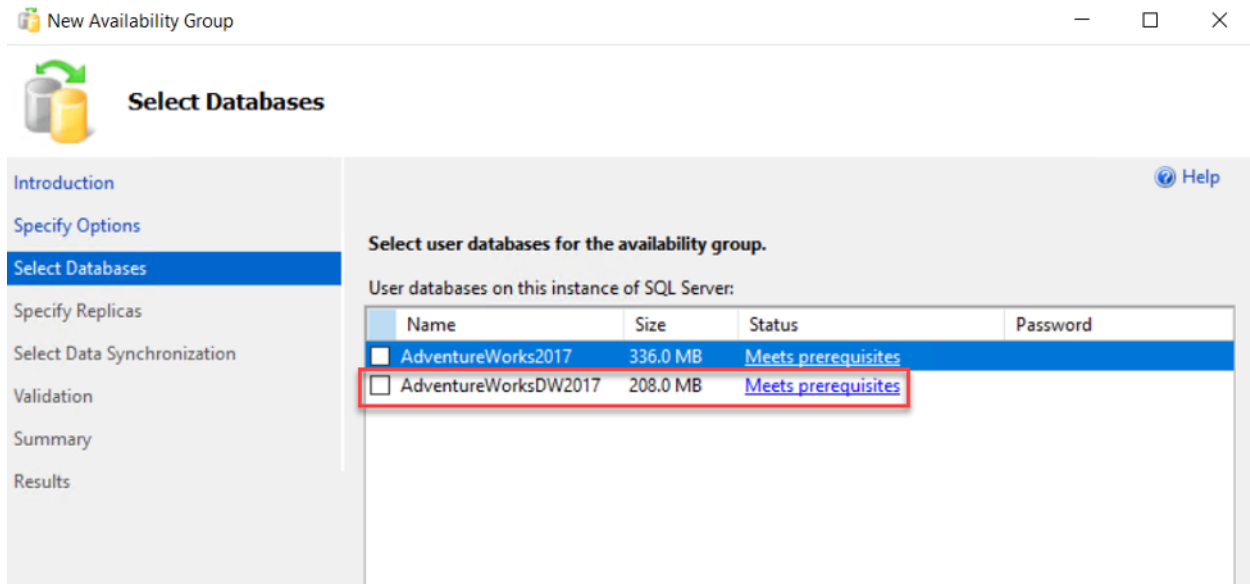
C:\Program Files\Microsoft SQL Server\MSSQL14.MSSQLSERVER\MSSQL\Backup\AdventureWorksDW2017.bak

Add... Remove Contents

Ready

OK Cancel

Refreshing the Select Databases window, the database status will be changed to Meet Prerequisites, as shown below:



## Shared Network Location Permission Issue

While trying to configure an Always on Availability Group site, the validation step of the New Availability Group wizard failed with the error message below:

*The primary server 'SQL1' cannot write to '\\SQL1\\AlwaysON\\BackupLocDb\_dbb55cb4-af89-4ed3-b189-1fcaad42358c.bak'. (Microsoft.SqlServer.Management.HadrModel)*


*Backup failed for Server 'SQL1'. (Microsoft.SqlServer.SmoExtended)*

*Cannot open backup device '\\SQL1\\AlwaysON\\BackupLocDb\_dbb55cb4-af89-4ed3-b189-1fcaad42358c.bak'. Operating system error 5(Access is denied.).*

*BACKUP DATABASE is terminating abnormally. (.Net SqlClient Data Provider)*

In the Full database and log backup initial synchronization method, a shared folder is required to keep the full backup and transaction log backup files temporarily to restore it to all secondary replicas. If the Primary replica is not able to write the backup files to it, or the secondary replicas are not able to read the backup files from it, the New Availability Group validation process will fail as below:

New Availability Group


 **Validation**

Introduction  
Specify Options  
Select Databases  
Specify Replicas  
Select Data Synchronization  
**Validation**  
Summary  
Results

**Results of availability group validation.**

Name	Result
✓ Checking for free disk space on the server instance that hosts secondary replica SQL2	<a href="#">Success</a>
✓ Checking if the selected databases already exist on the server instance that hosts second...	<a href="#">Success</a>
✓ Checking for the existence of the database files on the server instance that hosts seconda...	<a href="#">Success</a>
✓ Checking for compatibility of the database file locations on the server instance that host...	<a href="#">Success</a>
✓ Checking whether the endpoint is encrypted using a compatible algorithm	<a href="#">Success</a>
✗ Checking shared network location	<a href="#">Error</a>
✓ Checking replica availability mode	<a href="#">Success</a>

Microsoft SQL Server Management Studio

 The primary server 'SQL1' cannot write to  
'\\SQL1\\AlwaysON\\BackupLocDb\_dbb55cb4-af89-4ed3-b189-1fcaad42358c.bak'.  
(Microsoft.SqlServer.Management.HadrModel)

**Additional information:**

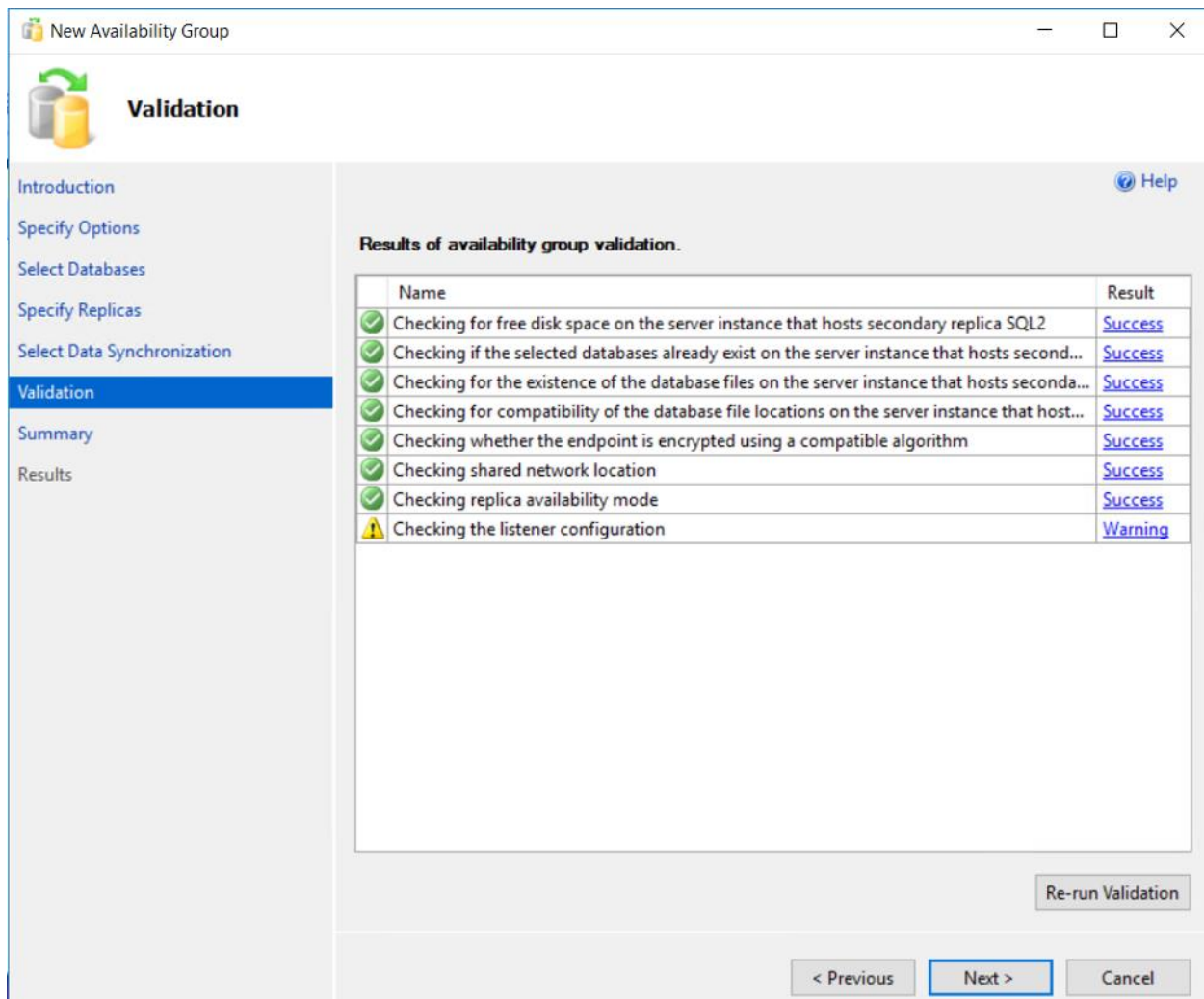
- ↳ Backup failed for Server 'SQL1'. (Microsoft.SqlServer.SmoExtended)
- ↳ An exception occurred while executing a Transact-SQL statement or batch.  
(Microsoft.SqlServer.ConnectionInfo)
- ↳ Cannot open backup device  
'\\SQL1\\AlwaysON\\BackupLocDb\_dbb55cb4-af89-4ed3-b189-1fcaad42358c.bak'. Operating  
system error 5(Access is denied.).  
BACKUP DATABASE is terminating abnormally. (Microsoft SQL Server, Error: 3201)

Help Copy message Show details OK

< Previous Next > Cancel

To fix that issue, we need to grant the SQL Server Service account of the Primary and Secondary replicas read and write permission on the shared folder shown in the error message, then re-run the validation process, to make sure that all checks are succeeded, as shown below:



































## Windows Failover Cluster Issue

Assume that you are checking the status of an existing Always on Availability Group site, and see that:

- The Primary role is moved from SQL1 instance to SQL2.
- In SQL2, the databases are in the Synchronized state.
- In SQL1, the databases are not synchronized.
- SQL1 is in Resolving state.

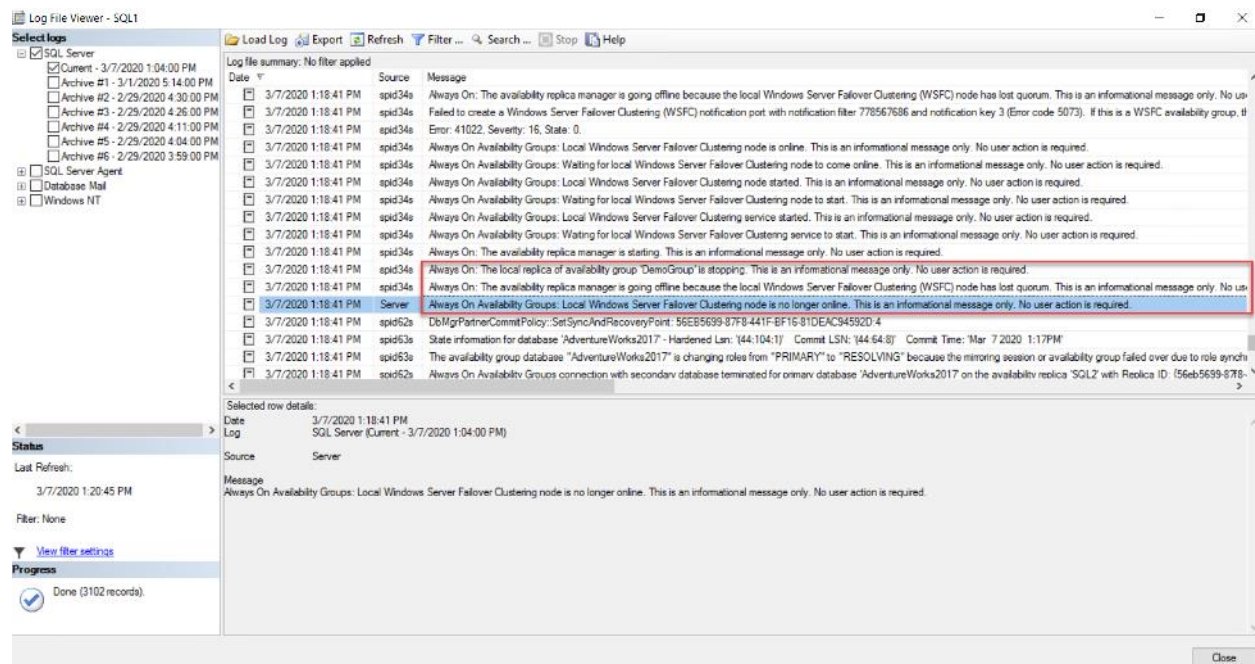
As you can see clearly from the SSMS Object Explorer below:

- [-]  SQL1 (SQL Server 14.0.3281.6 - AY\sqladmin)
  - [-]  **Databases**
    - [+]  System Databases
    - [+]  Database Snapshots
    - [+]  **AdventureWorks2017 (Not Synchronizing)**
    - [+]  AdventureWorksDW2017
  - [+]  Security
  - [+]  Server Objects
  - [+]  Replication
  - [+]  PolyBase
  - [-]  Always On High Availability
    - [-]  Availability Groups
      - [+]  **DemoGroup (Resolving)**
  - [+]  Management
  - [+]  Integration Services Catalogs
  - [+]  SQL Server Agent
  - [+]  XEvent Profiler
- [-]  SQL2 (SQL Server 14.0.3281.6 - AY\sqladmin)
  - [-]  Databases
    - [+]  System Databases
    - [+]  Database Snapshots
    - [+]  **AdventureWorks2017 (Synchronized)**
  - [+]  Security
  - [+]  Server Objects
  - [+]  Replication
  - [+]  PolyBase
  - [-]  Always On High Availability
    - [-]  Availability Groups
      - [+]  **DemoGroup (Primary)**
  - [+]  Management



Checking the SQL Server Error logs in the problematic node, we can see that the Availability Group replica becomes offline and the Availability Group stopped working due to an issue in the Windows Server Failover Cluster, as shown in the errors below:

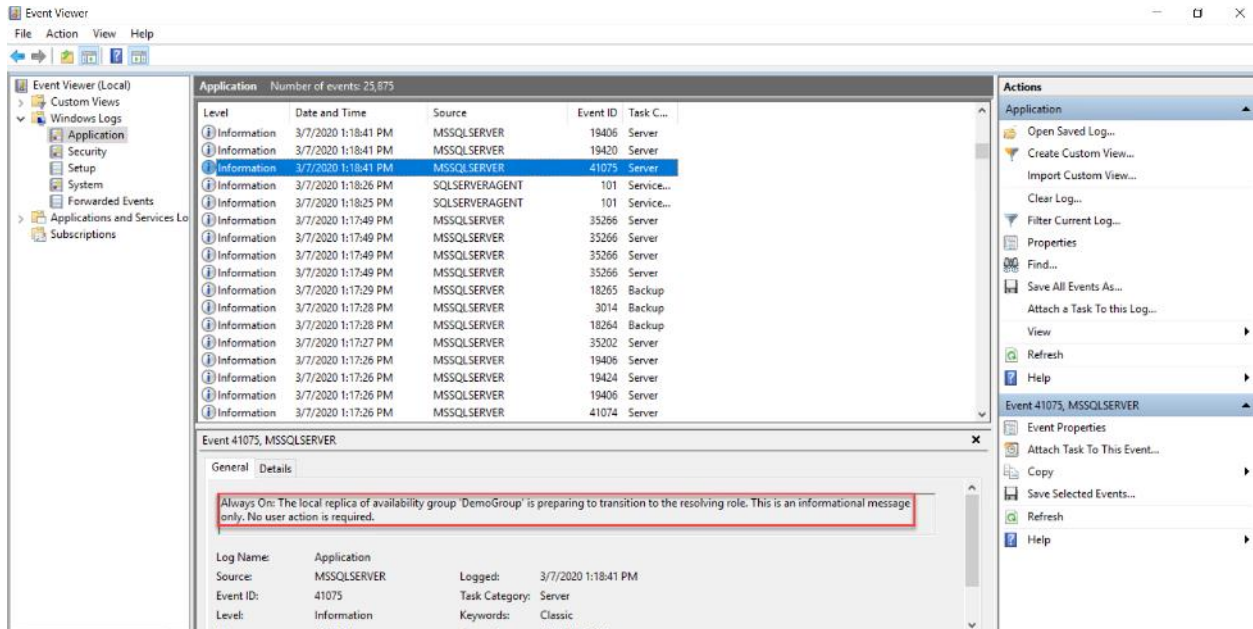
- **Always On Availability Groups: Local Windows Server Failover Clustering node is no longer online.** This is an informational message only. No user action is required.
- **Always On: The availability replica manager is going offline because the local Windows Server Failover Clustering (WSFC) node has lost quorum.** This is an informational message only. No user action is required.
- **Always On: The local replica of the availability group 'DemoGroup' is stopping.** This is an informational message only. No user action is required.



The same thing can be detected from the Windows Server Event Viewer, that shows gradually how the replica changes its state to Resolving state, as below:

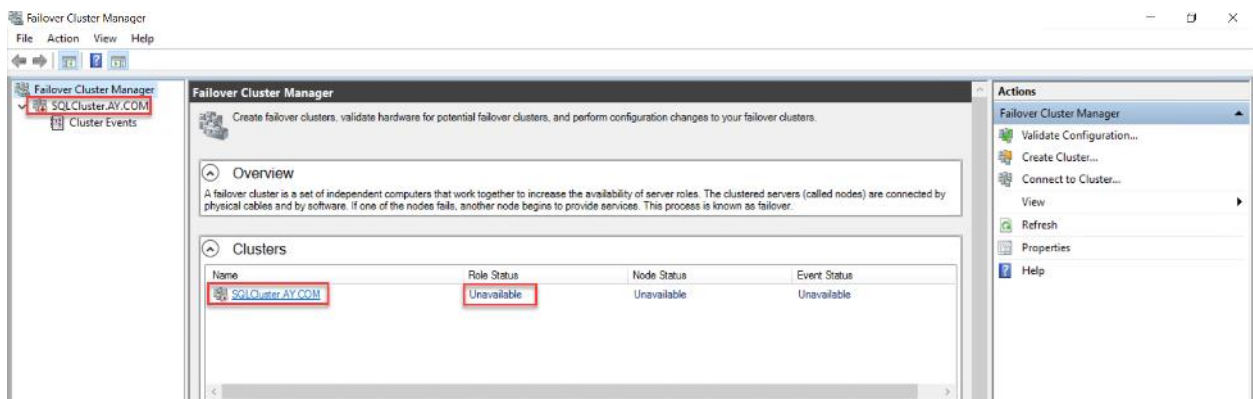
- **Always On: The local replica of availability group 'DemoGroup' is preparing to transition to the resolving role.** This is an informational message only. No user action is required.
- **The availability group 'DemoGroup' is being asked to stop the lease renewal because the availability group is going offline.** This is an informational message only. No user action is required.
- **The state of the local availability replica in availability group 'DemoGroup' has changed from 'PRIMARY\_NORMAL' to 'RESOLVING\_NORMAL'.** The state changed

*because the availability group is going offline. The replica is going offline because the associated availability group has been deleted, or the user has taken the associated availability group offline in Windows Server Failover Clustering (WSFC) management console, or the availability group is failing over to another SQL Server instance. For more information, see the SQL Server error log or cluster log. If this is a Windows Server Failover Clustering (WSFC) availability group, you can also see the WSFC management console.*

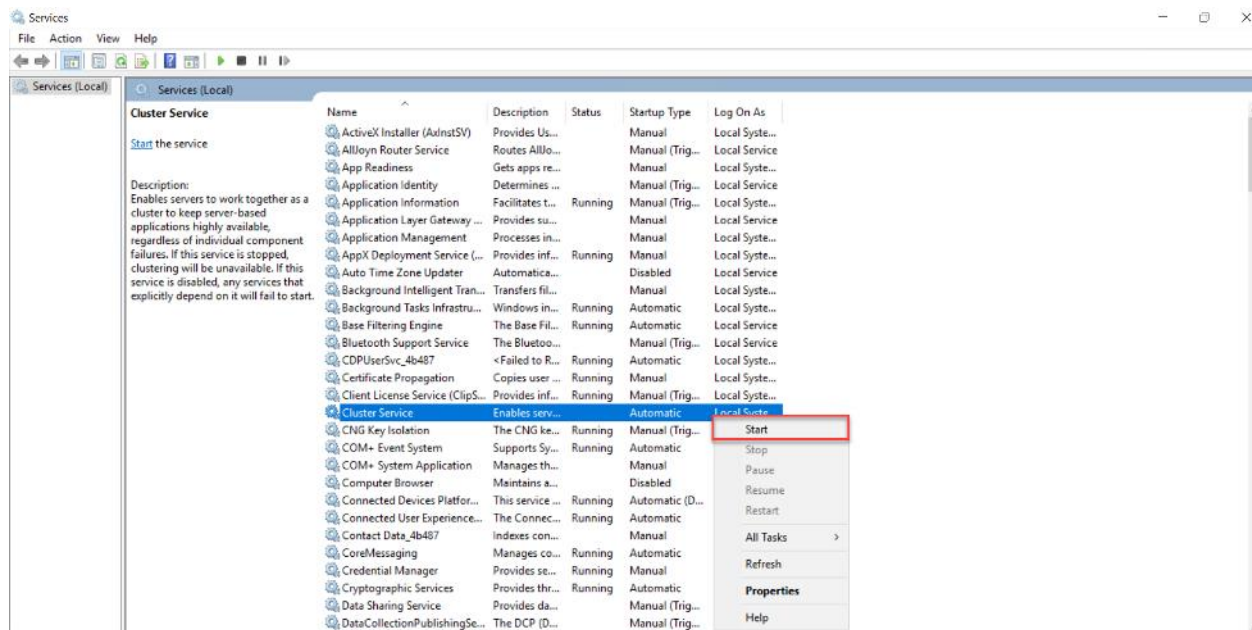


To check the status of the Windows Cluster site, we will use the Failover Cluster Manager to see which part of the Windows Cluster is failing.

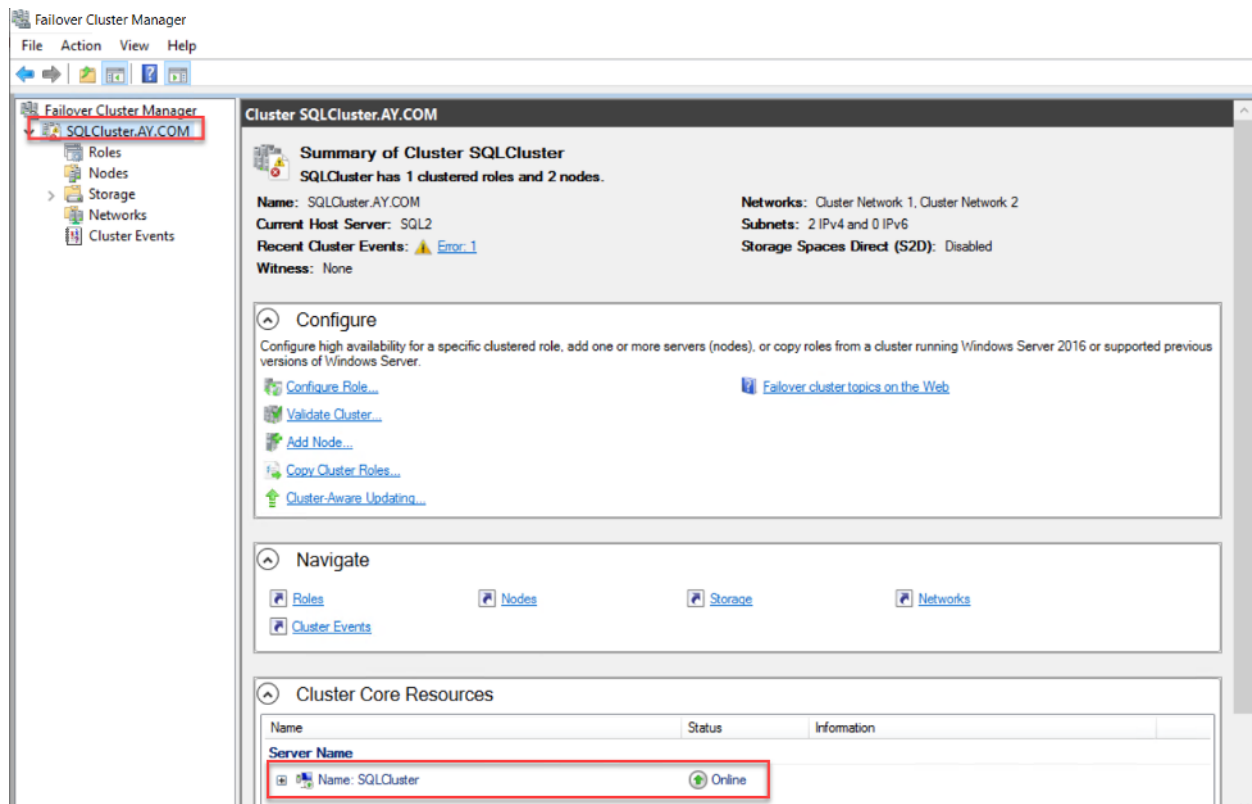
But the Failover Cluster Manager shows that the whole cluster is down, as shown below:



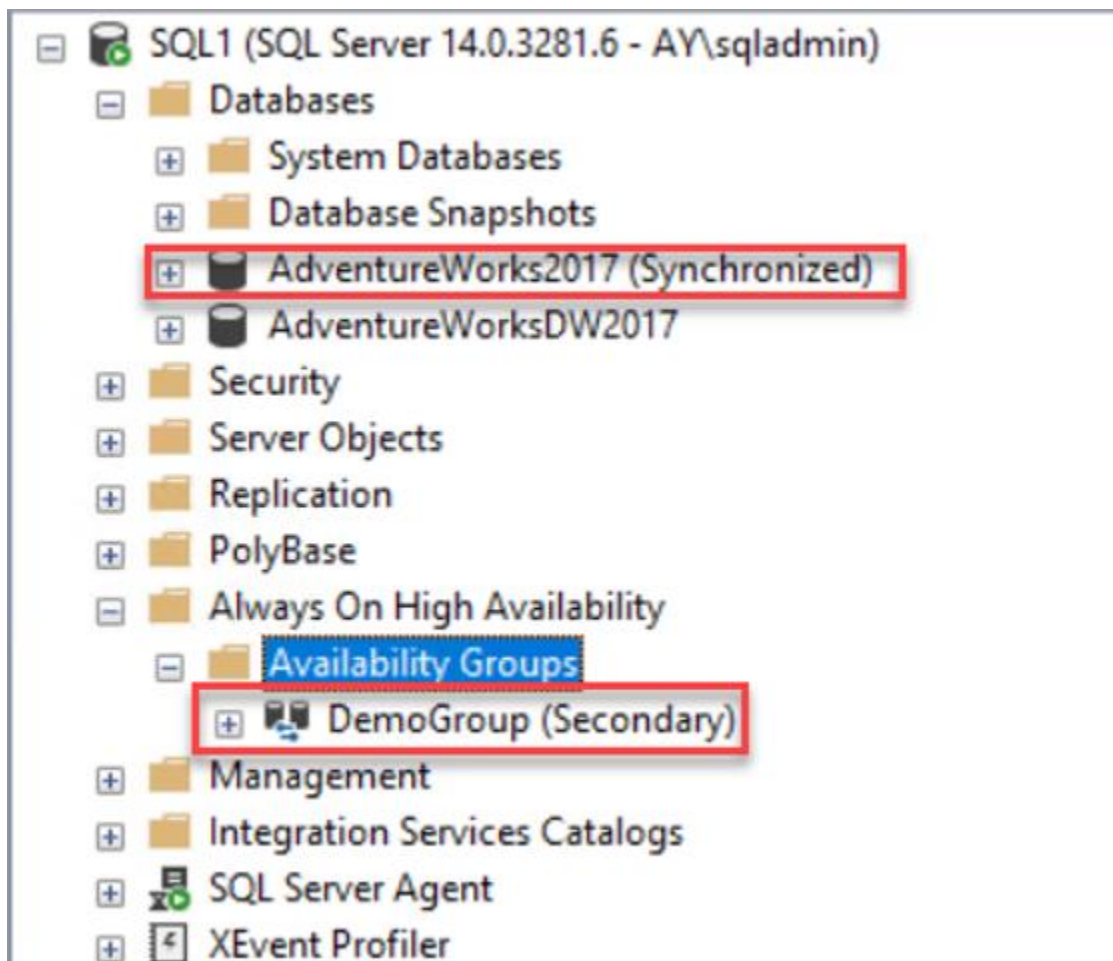
The first thing to validate here from the Windows Failover Cluster side is the Cluster Service, that can be checked from the Windows Services console, as below:



It is clear from the Services console, that the Cluster Service is not running. To fix that issue, start the service from that console, then refresh the Failover Cluster Manager console to make sure that the Windows Cluster site is up and running, as shown below:

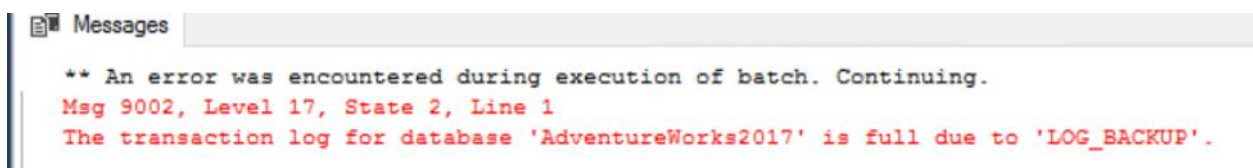


Checking the Always on Availability Group again, you will see that the databases are in synchronized again and the Always on Availability Group site is in health state again, as shown below:



## Transaction Log File is Full at the Primary Side

Assume that you receive the below error message when trying to execute a new query on one of the Always on Availability Group databases:



Checking what is blocking the transaction logs file and prevents it from being truncated, you will see that the transaction log file of this database is pending log backup operation to be truncated, as shown below:

```
SELECT name ,log_reuse_wait_desc FROM sys.databases
```

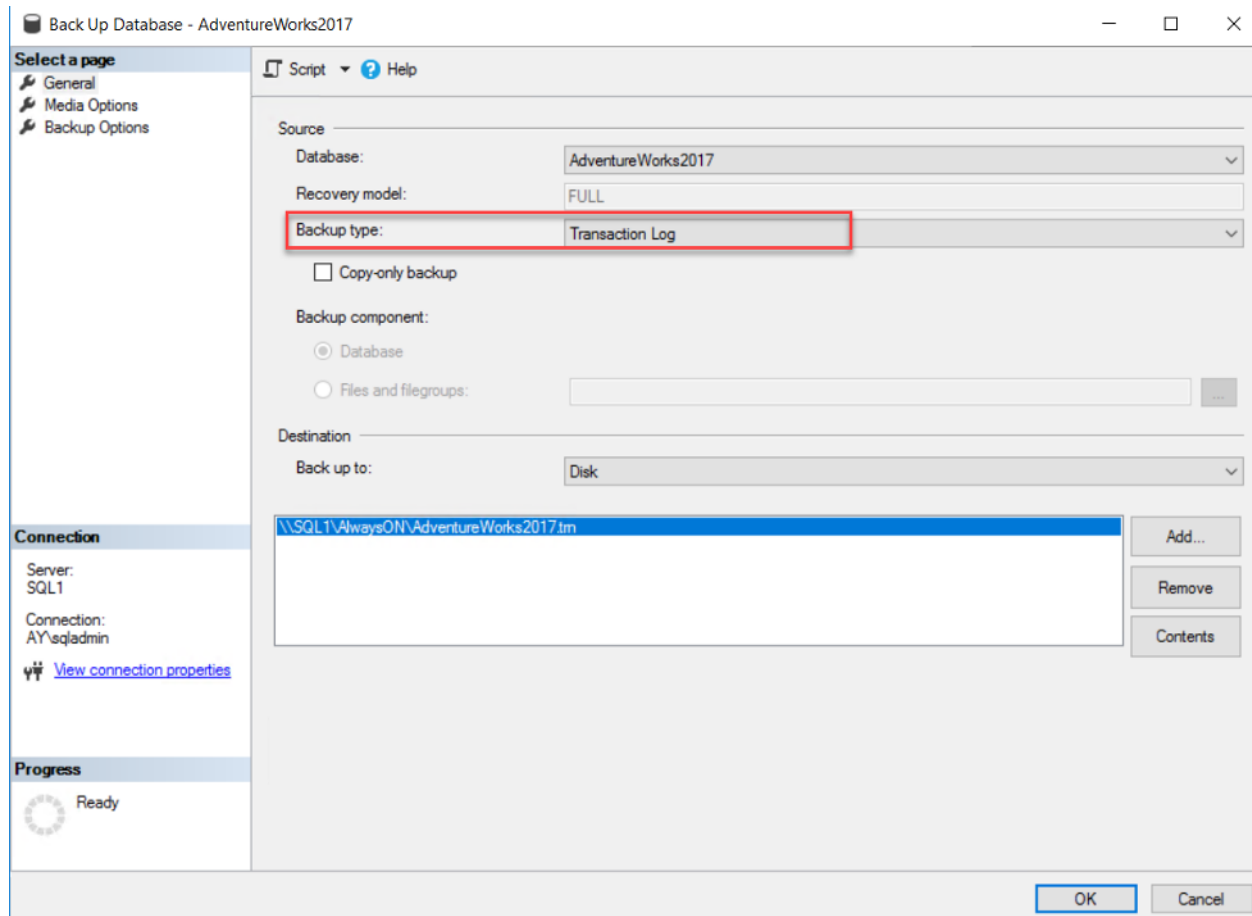
100 %

Results Messages

	name	log_reuse_wait_desc
1	master	NOTHING
2	tempdb	NOTHING
3	model	NOTHING
4	msdb	NOTHING
5	AdventureWorks2017	LOG_BACKUP
6	AdventureWorksDW2017	NOTHING
7	distribution	NOTHING

Taking a transaction log backup for that database, in case you forget to schedule a transaction log backup job, as follows:





And check again what is blocking the transaction log of that database, it is showing in my scenario that, it is waiting for Availability\_Replica. Which means that the logs are waiting to be written to the secondary replica, but not able to send these transaction logs to the secondary replicas due to an issue in the Always on Availability Group site, as below:



```
SELECT name ,log_reuse_wait_desc FROM sys.databases
```



100 %


Results Messages

	name	log_reuse_wait_desc
1	master	NOTHING
2	tempdb	NOTHING
3	model	NOTHING
4	msdb	NOTHING
5	AdventureWorks2017	AVAILABILITY_REPLICA
6	AdventureWorksDW2017	NOTHING
7	distribution	NOTHING

The best location to check and troubleshoot the Always on Availability Group site is the Always on Dashboard, which can be opened by right-clicking on the Availability Group name and choose Show Dashboard option.



From the dashboard, you can see that the Secondary replica SQL2 is not synchronized with the Primary replica, due to connectivity issue, as shown below:


 DemoGroup: hosted by SQL1 (Replica role: Primary) Last updated: 3/7/2020 1:49:34 PM  
Auto refresh: on 



Availability group state:  [Critical --- Critical \(1\), Warnings \(3\)](#) [Start Failover Wizard](#)  
[View Always On Health Events](#)  
[View Cluster Quorum Information](#)

Primary instance: SQL1  
Failover mode: Automatic  
Cluster state: SQLCluster (Normal Quorum)  
Cluster type: Windows Server Failover Cluster [Collect Latency Data](#)  
[Analyze Log Block Latency](#)  
[Add/Remove Columns](#)

Availability replica:



Name	Role	Availability Mode	Failover Mode	Seeding Mode	Synchronization State	Issues
 SQL1	Primary	Synchronous co...	Automatic	Manual	Synchronized	
 SQL2	Secondary	Synchronous co...	Automatic	Manual	Not Synchronizing	<a href="#">Critical ...</a>

Group by ▾ [Add/Remove Columns](#) 

Name	Replica	Synchronization State	Failover Read...	Issues
SQL1				
 AdventureWorks2017	SQL1	Synchronized	No Data Loss	
 AdventureWorks2017	SQL2	Not Synchronizing	Data Loss	<a href="#">Warnings (1)</a>

Policy Evaluation Result on Availability Replica SQL2

Select a policy to see the detailed results in the table below

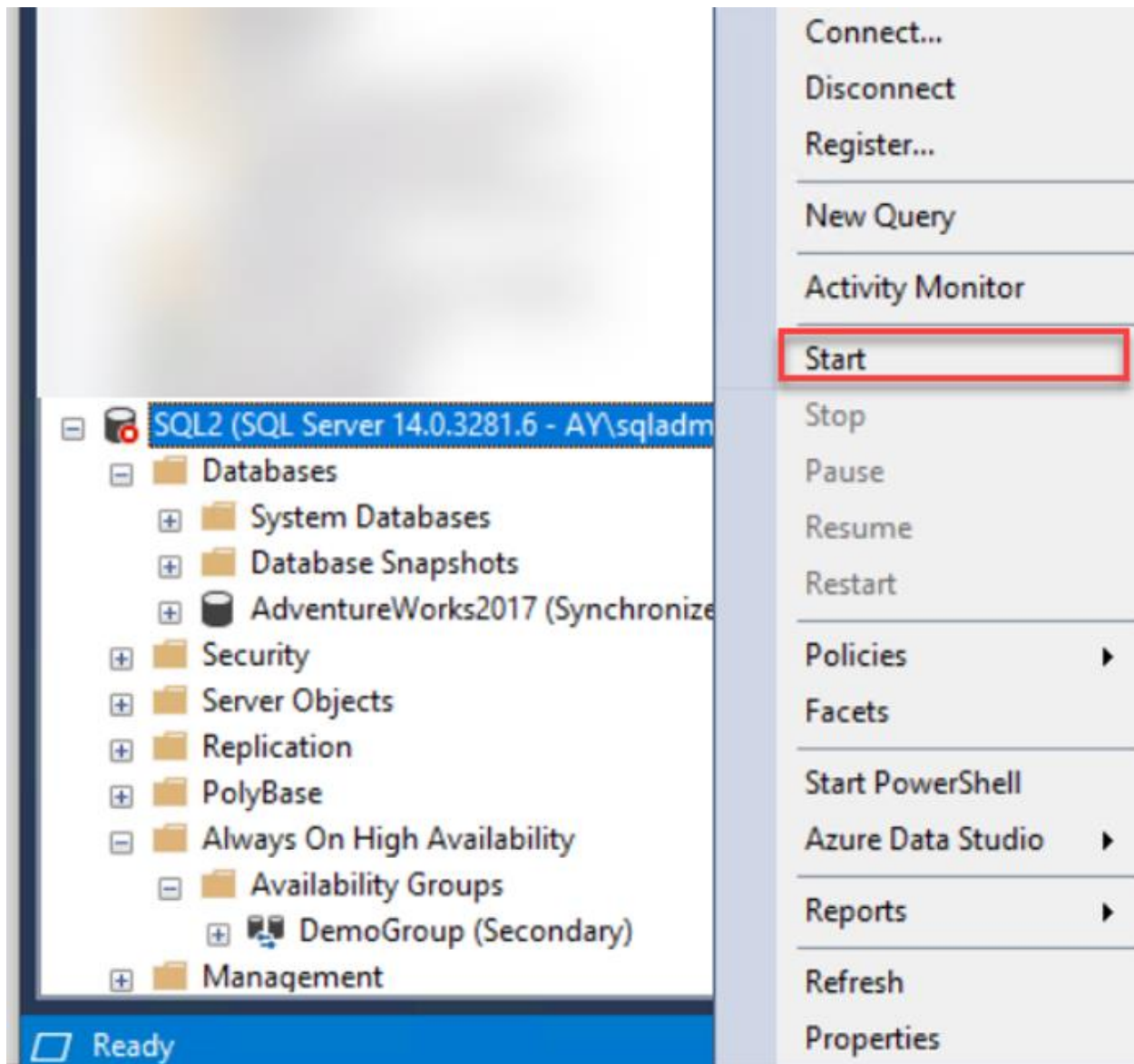
Detected Issue
 Availability replica is disconnected.
 Data synchronization state of some availability database is not healthy.

Description: [More Information](#)



This secondary replica is not connected to the primary replica. The connected state is DISCONNECTED.


Close Help

Checking the Secondary replica, and making sure that the SQL Server Service is up and running at the secondary side, as follows:





Then refreshing the Availability Group dashboard again, you will see that the Always on Availability Group site is healthy again. Checking if the transaction logs file is blocked by any operation, we will see that it is pending OLDEST\_PAGE, indicating that the oldest page of the database is older than the checkpoint LSN. This issue can be fixed easily by taking another transaction log backup and the transaction log file will be blocked by nothing, as shown clearly below:


 DemoGroup: hosted by SQL1 (Replica role: Primary)
 Last updated: 3/7/2020 1:53:03 PM  
 Auto refresh: on 



Availability group state:  Healthy  
 Primary instance: SQL1  
 Failover mode: Automatic  
 Cluster state: SQLCluster (Normal Quorum)  
 Cluster type: Windows Server Failover Cluster

[Start Failover Wizard](#)  
[View Always On Health Events](#)  
[View Cluster Quorum Information](#)  
[Collect Latency Data](#)  
[Analyze Log Block Latency](#)  
[Add/Remove Columns](#)

Availability replica:

Name	Role	Availability Mode	Failover Mode	Seeding Mode	Synchronization State	Issues
 SQL1	Primary	Synchronous co...	Automatic	Manual	Synchronized	
 SQL2	Secondary	Synchronous co...	Automatic	Manual	Synchronized	

Group by ▾ [Add/Remove Columns](#) 

Name	Replica	Synchronization State	Failover Read...	Issues
SQL1				
 AdventureWorks2017	SQL1	Synchronized	No Data Loss	
SQL2				
 AdventureWorks2017	SQL2	Synchronized	No Data Loss	

```
SELECT name ,log_reuse_wait_desc FROM sys.databases
```

00 % ▾

Results Messages

	name	log_reuse_wait_desc
1	master	NOTHING
2	tempdb	NOTHING
3	model	NOTHING
4	msdb	NOTHING
5	AdventureWorks2017	OLDEST_PAGE
6	AdventureWorksDW2017	NOTHING
7	distribution	NOTHING

After the next TRN backup

```
SELECT name ,log_reuse_wait_desc FROM sys.databases
```

100 % ▾

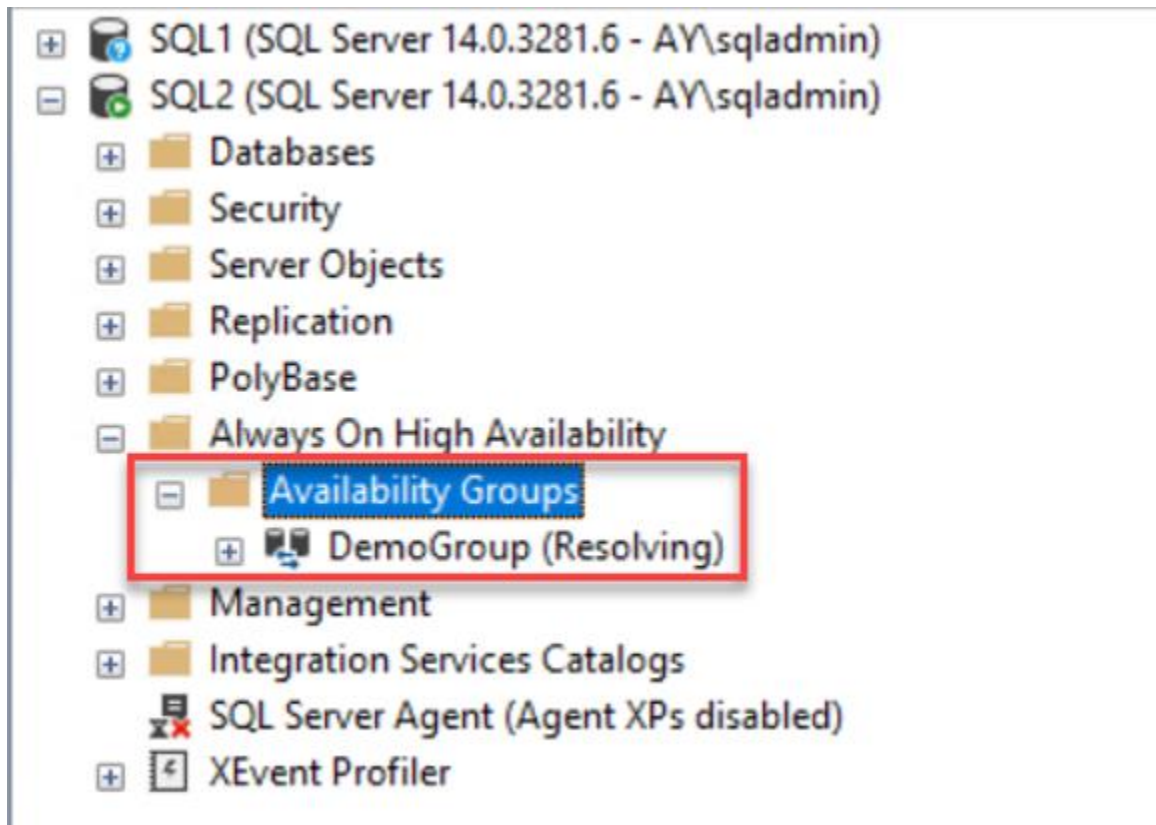
Results Messages

	name	log_reuse_wait_desc
1	master	NOTHING
2	tempdb	NOTHING
3	model	NOTHING
4	msdb	NOTHING
5	AdventureWorks2017	NOTHING
6	AdventureWorksDW2017	NOTHING
7	distribution	NOTHING

## Always on Availability Group Failover Misconfiguration

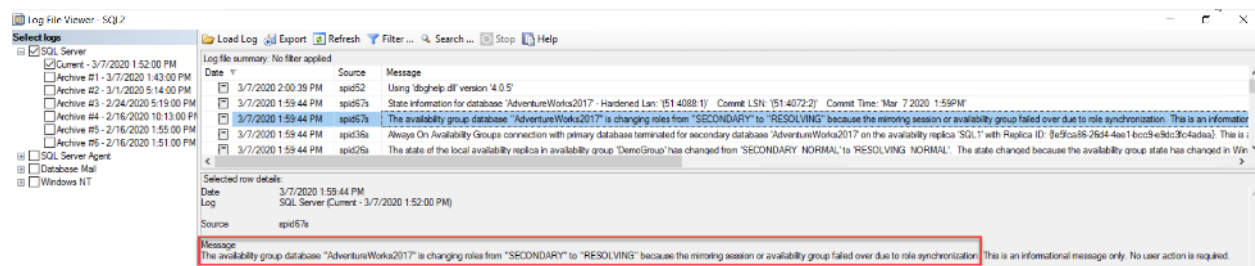
Assume that the Primary replica becomes offline due to an unplanned issue. As expected, the system will not be affected as an automatic failover operation will be performed and the secondary replica will act as the new Primary replica.

But in our case, this happy scenario is not valid, where the secondary replica changed to Resolving state and the system is down!



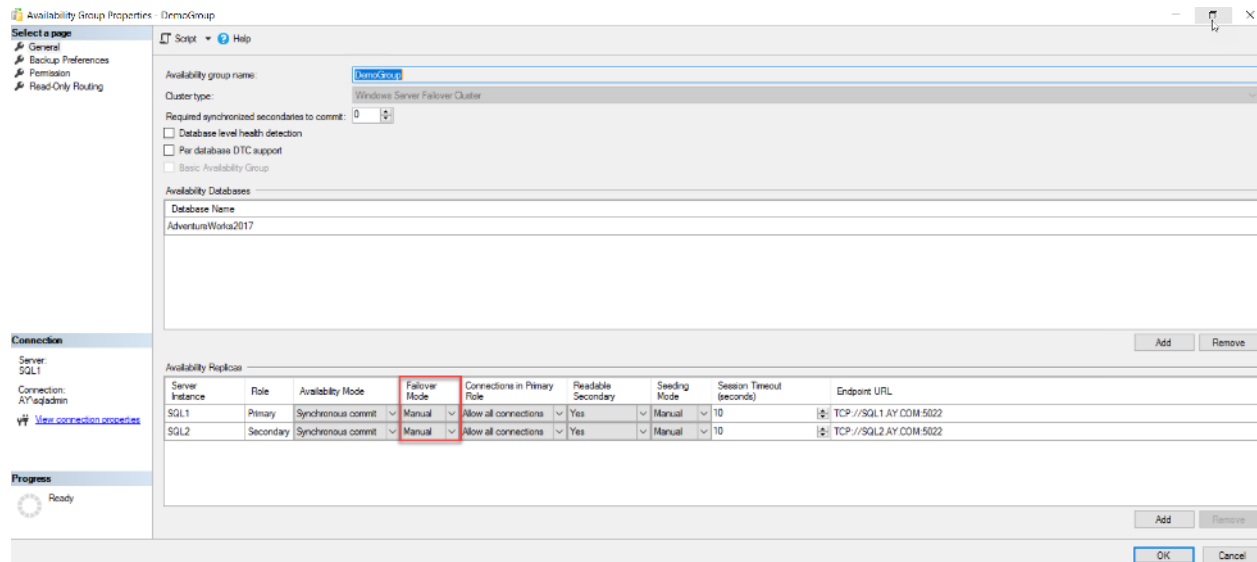
Checking the secondary replica's error log and see why it is not acting as the new Primary as expected, you will see that it is failing due to a role synchronization issue, as shown below:

*The availability group database "AdventureWorks2017" is changing roles from "SECONDARY" to "RESOLVING" because the mirroring session or availability group failed over due to role synchronization. This is an informational message only. No user action is required.*

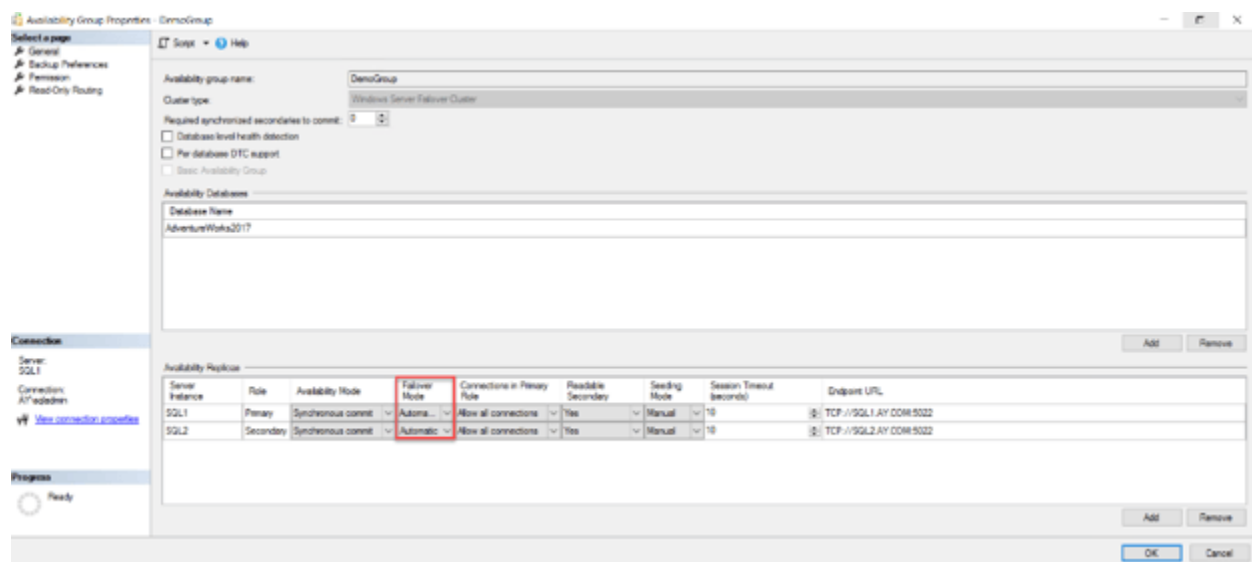


This means that there is an issue with the synchronization mode that is used in this Availability Group. The synchronization mode used, can be checked from the Always on Availability Group properties page.

From the properties page below, it is clear that the Failover mode in this Availability Group is configured to be performed Manually only. In this case, you need to manually perform a failover operation before rebooting or shutting down the server:



This can be fixed easily by changing the Failover Mode to Automatic, where an automatic failover operation will be performed in case of any unplanned shutdown or reboot:



The same issue can be faced when the Windows Failover Cluster quorum is configured with Node Majority for an even number of replicas, where any failure for one of the servers will bring the Windows Failover Cluster site offline. For more information, check [Windows Failover Cluster Quorum Modes in SQL Server Always On Availability Groups](#):



The screenshot displays the SQL Server Enterprise Manager interface. At the top, a status bar indicates 'DemoGroup: hosted by SQL1 (Replica role: Primary)' and 'Last updated: 3/7/2020 2:37:24 PM'. The main pane shows the 'Availability group state' as 'Healthy'. Below this, a list of 'Availability replica' is shown with columns for Name, Role, and Availability. The replicas are SQL1 (Primary, Synchronizing) and SQL2 (Secondary, Synchronizing). A 'Cluster Quorum Information' window is open, showing the 'Cluster Name' as 'SQLCluster' and the 'Quorum Model' as 'Node Majority'. A red arrow points from the 'View Cluster Quorum Information' link in the right-hand pane to the 'Cluster Quorum Information' window. The window contains a table with the following data:

Member Name	Member Type	Member State	Vote Count
SQL1	Node	Online	1
SQL2	Node	Online	1

At the bottom of the window, there is a link: 'Click here for more information about configuring a WSFC cluster for Always On Availability Groups.' and a 'Close' button.

## Failover with Data Loss

Assume that you are trying to perform a manual failover between the Primary and one of the Secondary replicas, but in the Select New Primary Replica window, you see a warning message that the failover operation may end up with data loss as the Primary and the selected Secondary replica are not synchronized, as shown below:

Fail Over Availability Group: DemoGroup

## Select New Primary Replica

Introduction  
Select New Primary Replica  
Confirm Potential Data Loss  
Connect to Replica  
Summary  
Results

Select the new primary replica for this availability group.

Current Primary Replica: SQL1  
Primary Replica Status: Synchronous commit and Online  
Quorum Status: Normal Quorum

Choose new primary replica:

Server Instance	Availability Mode	Failover Mode	Failover Readiness	Warnings	Role
SQL2	Synchronous co...	Automatic	Data loss, Warnings(1)		Secondary

Microsoft SQL Server Management Studio


This replica has one or more databases that are not synchronized. Failing over to this replica could result in data loss for any transactions that did not reach this secondary replica prior to failing over.


Copy message OK

Refresh

< Previous Next > Cancel



To identify the cause of that issue, we will browse the Always on Health events using the Always on Availability Group dashboard, which shows that the Primary replica is not able to open a connection to the Secondary replica, as shown below:

 DemoGroup: hosted by SQL1 (Replica role: Primary)
 Last updated: 3/7/2020 2:32:21 PM



Availability group state:  Healthy  
 Primary instance: SQL1  
 Failover mode: Automatic  
 Cluster state: SQLCluster (Normal Quorum)  
 Cluster type: Windows Server Failover Cluster

[Start Failover Wizard](#)  
[View Always On Health Events](#)  
[View Cluster Quorum Information](#)  
[Collect Latency Data](#)  
[Analyze Log Block Latency](#)  
[Add/Remove Columns](#)

Availability replica:

Name	Role	Availability Mode	Failover Mode	Seeding Mode	Synchronization State	Issues
 SQL1	Primary	Synchronous co...	Automatic	Manual	Synchronized	
 SQL2	Secondary	Synchronous co...	Automatic	Manual	Synchronized	

Group by ▾

Name	Replica	Synchronization State	Failover Read...	Issues
SQL1				
 AdventureWorks2017	SQL1	Synchronized	No Data Loss	
SQL2				
 AdventureWorks2017	SQL2	Synchronized	No Data Loss	

[Add/Remove Columns](#)

Displaying 2543 Events

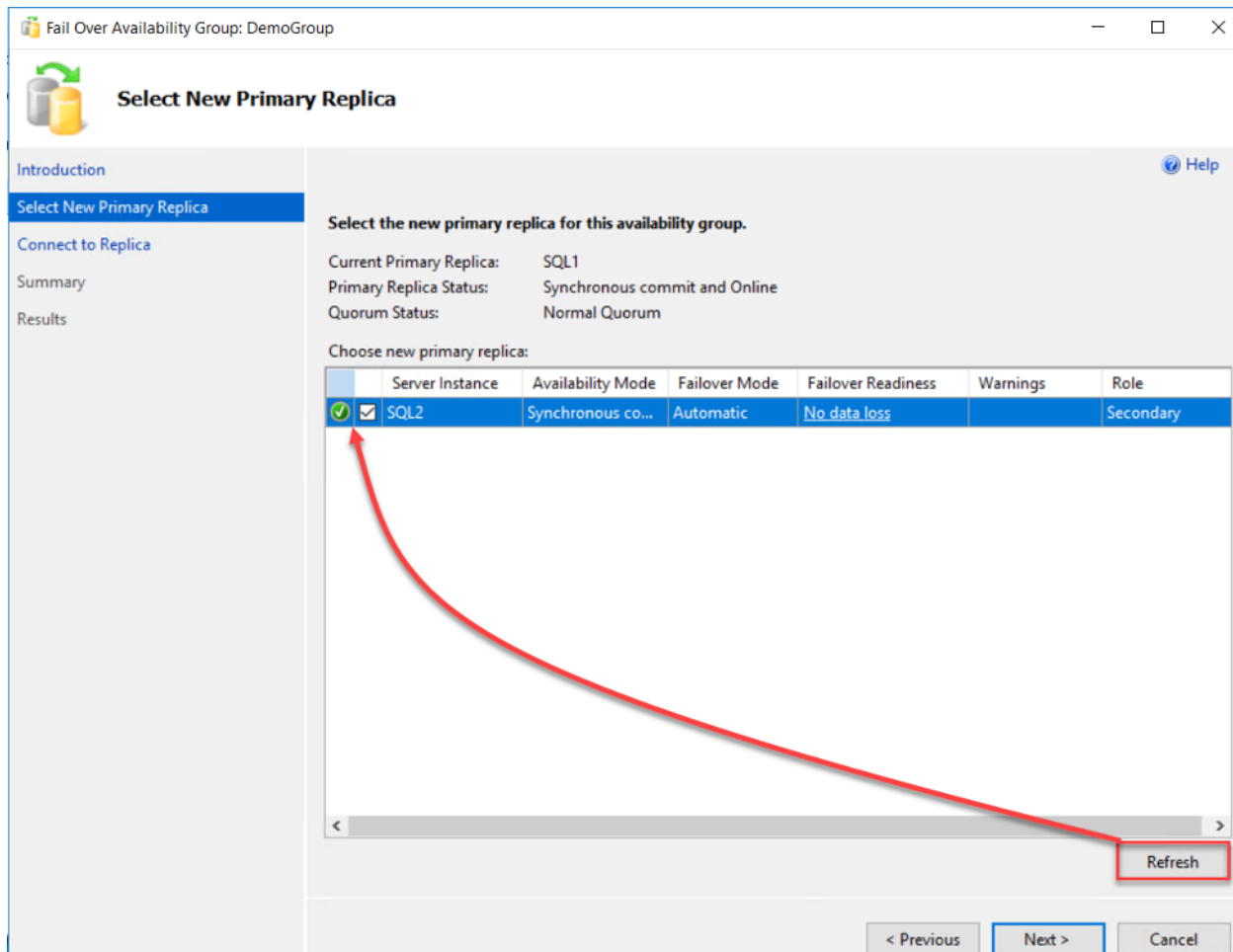
name	timestamp
hadr_db_partner_set_sync_st...	2020-03-07 14:07:54.4543854
alwayson_ddl_executed	2020-03-07 14:09:06.9095225
alwayson_ddl_executed	2020-03-07 14:09:06.9160084
alwayson_ddl_executed	2020-03-07 14:09:06.9900981
alwayson_ddl_executed	2020-03-07 14:09:07.0020030
error_reported	2020-03-07 14:09:31.0997181
hadr_db_partner_set_sync_st...	2020-03-07 14:09:31.8126416
error_reported	2020-03-07 14:30:42.7314543
hadr_db_partner_set_sync_st...	2020-03-07 14:30:44.8034496
hadr_db_partner_set_sync_st...	2020-03-07 14:30:44.8071848
hadr_db_partner_set_sync_st...	2020-03-07 14:30:44.8189128

Event:error\_reported (2020-03-07 14:09:31.0997181)

Details

Field	Value
category	SERVER
destination	ERRORLOG, EVENTLOG
error_number	35206
is_intercepted	False
message	A connection timeout has occurred on a previously established connection to availability replica 'SQL2' with id [56EB5699-87F8-441F-BF16-81DEAC94...
severity	10
state	1
user_defined	False

After fixing the connectivity issue between the Primary and the Secondary, refresh the replicas list and you will see that the data loss issue is fixed, as shown below. For more information about troubleshooting the connectivity issues, check [Troubleshoot connecting to the SQL Server Database Engine](#).



## Monitoring Always on Availability Group Latency

The Availability Group dashboard can be modified to include additional columns that provide information about the synchronization latency between Primary and Secondary replicas, including the Commit LSN, Sent LSN and harden LSN values, without showing why there is a latency, as shown below:

**DemoGroup: hosted by SQL1 (Replica role: Primary)**

Availability group state: ✔ Healthy  
 Primary instance: SQL1  
 Failover mode: Automatic  
 Cluster state: SQLCluster (Normal Quorum)  
 Cluster type: Windows Server Failover Cluster

Availability replica:

Name	Role	Availability Mode	Failover Mode	Seeding Mode	Synchronization State	Issues
<span style="color: green;">✔</span> SQL1	Primary	Synchronous co...	Automatic	Manual	Synchronized	
<span style="color: green;">✔</span> SQL2	Secon...	Synchronous co...	Automatic	Manual	Synchronized	

Group by ▾

Name	Replica	Synchronization State	Failover Read...	Issues
SQL1				
<span style="color: green;">✔</span> AdventureWorks2017	SQL1	Synchronized	No Data Loss	
SQL2				
<span style="color: green;">✔</span> AdventureWorks2017	SQL2	Synchronized	No Data Loss	

Right-hand pane metrics:

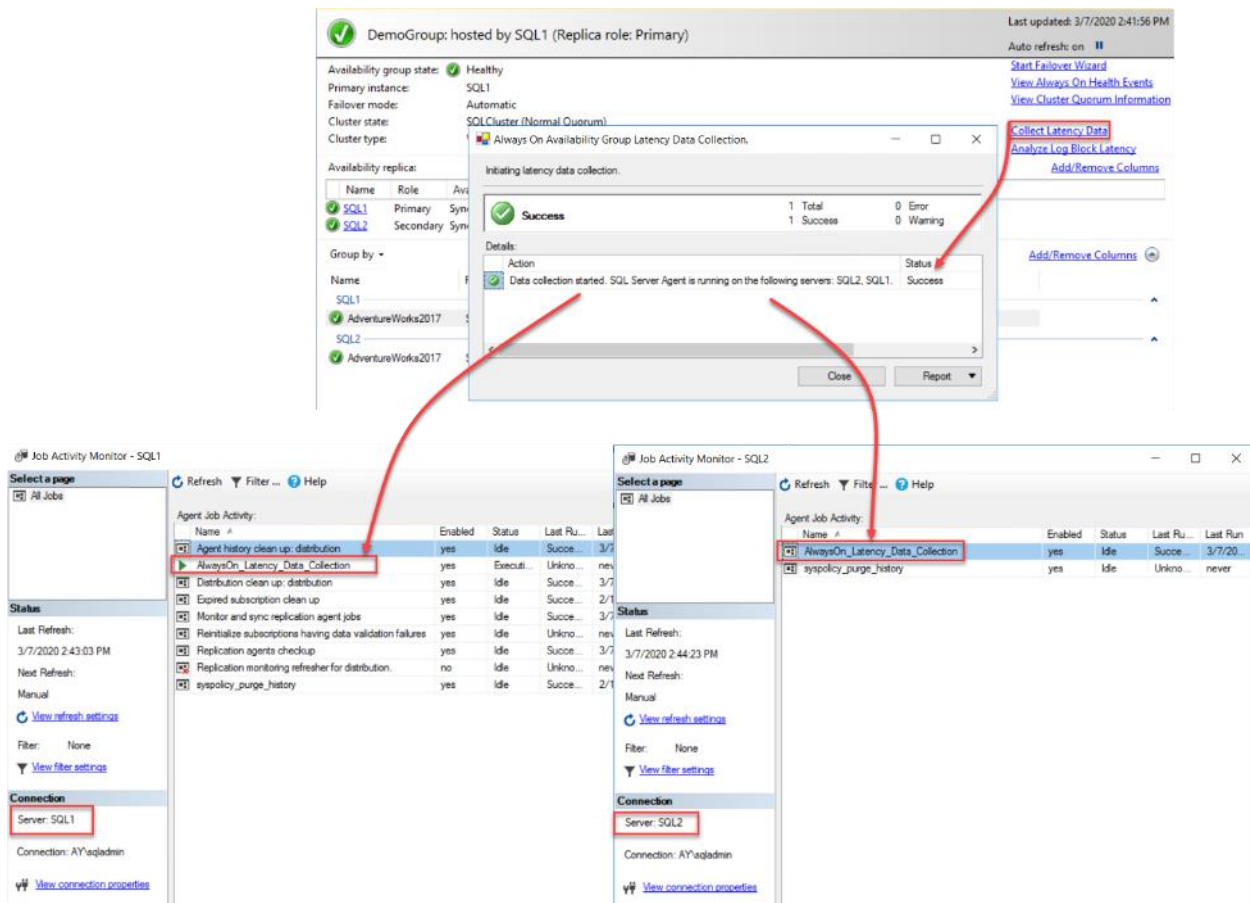
- Name
- Replica
- Synchronization State
- Failover Readiness
- Issues
- Suspended
- Suspend Reason
- Estimated Recovery Time (seconds)
- Estimated Data Loss (time)
- Synchronization Performance (seconds)
- Log Send Queue Size (KB)
- Log Send Rate (KB/sec)
- Redo Queue Size (KB)
- Redo Rate (KB/sec)
- FileStream Send Rate (KB/sec)
- End of Log LSN
- Recovery LSN
- Truncation LSN
- Last Commit LSN
- Last Commit Time
- Last Sent LSN
- Last Sent Time
- Last Received LSN
- Last Received Time
- Last Hardened LSN
- Last Hardened Time
- Last Redone LSN
- Last Redone Time

For more information about measuring the latency, check the [Measuring Availability Group synchronization lag](#).

Starting from SSMS 17.4, the Always on Availability Group dashboard enhanced to include two new options that are used for latency information calculation, analysis and reporting, which helps in identifying the bottlenecks in the transaction logs flow between the Primary and the Secondary replicas and narrow down the cause of that latency.

For more information about the new functionality and reports, check to Use the [Always on Availability Group dashboard](#).

To trigger using this new option, click on **Collect Latency Data** option from the Always on Availability Group dashboard, that will create a new SQL Agent job on the Primary and Secondary replicas to collect the latency data, As shown below:



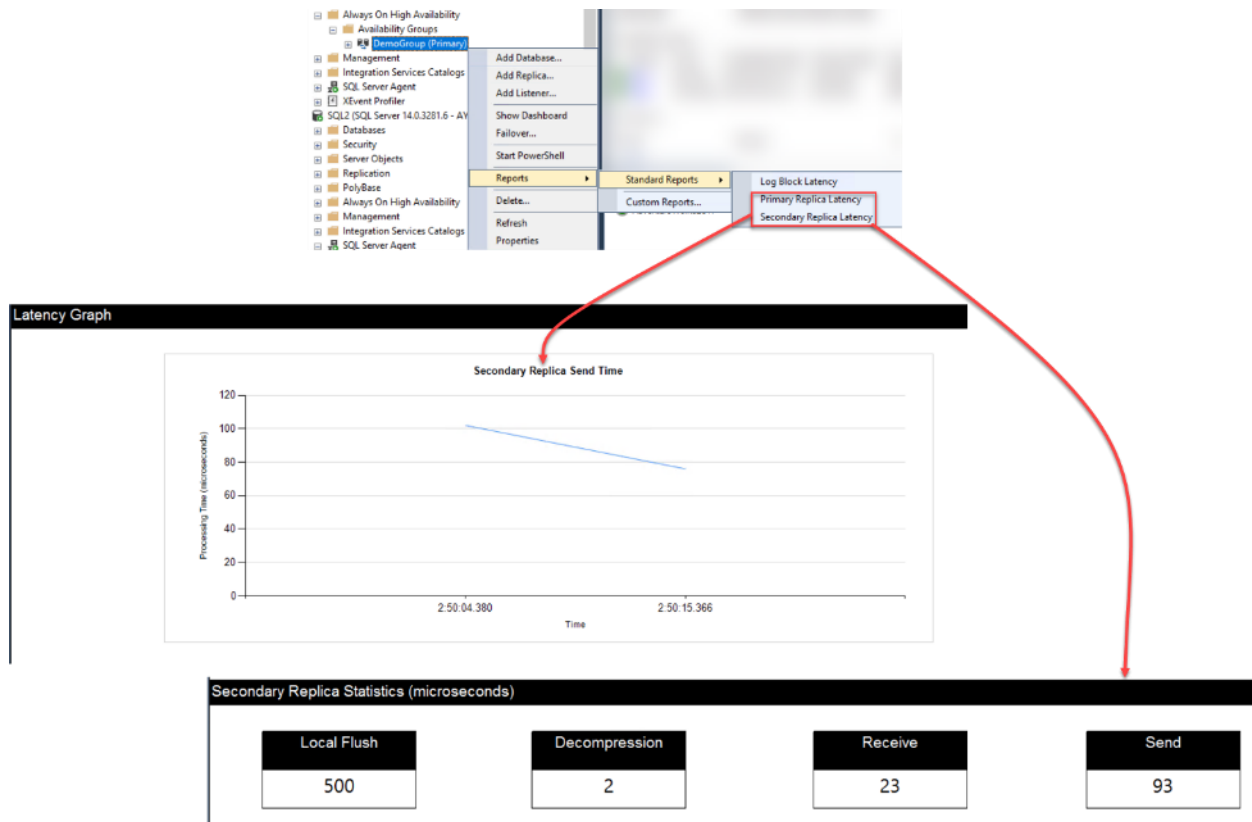
When the created job execution has completed on all the Availability Group replicas, you will be able to view the latency statistics from the latency reports by right-clicking on the Availability Group name and choose the Primary Replica Latency or Secondary Replica Latency report, based on the replica role in the Availability Group.

After providing information about the Availability Group replicas, the latency report will show a graphical view of the transaction log commit time on the Primary replica and the remote Hardening time for the secondary replicas, aggregated as average values. Also, the report provides statistical values for the transaction logs send, receive, commit, compress, decompress and other numerical values based on the replica role in the Availability Group.

For more information about the latency report, [check New in SSMS - Always On Availability Group Latency Reports](#).

The below report is an example of the latency reports generated from the Secondary replica, showing normal logs transport operations:





Also, the **Log Block Latency** report shows the amount of time, in ms, that the transaction log on the Primary replica waits for Secondary replicas to commit that transaction. After enabling it from the Availability Group Dashboard, you can browse it from the SSMS similar to the previous latency reports. Take into consideration that, the large latency time indicates that the Primary replica is waiting a long time for the Secondary replicas to commit the sent transactions, as shown below:

**Log block latency for synchronous commit replicas (ms)**

Database Name: AdventureWorks2017  
 Log block id: 219043336296  
 Start time: 03/07/2020 02:50:04.368 PM  
 Commit time: 03/07/2020 02:50:04.370 PM  
 Duration: 2

Availability Replica Name	Role	Log capture initialization time on primary for secondary	Log capture completion time on primary for secondary	Processing time to queue log message to network on primary for secondary	Log flush initialization time	Log flush completion time	Processing time to harden log message on secondary	Processing time to queue ack message to network on secondary	Round trip network latency between primary and secondary	Processing time to signal commit on primary
SQL1	PRIMARY	Not Applicable	Not Applicable	Not Applicable	0	1	Not Applicable	Not Applicable	Not Applicable	0
SQL2	SECONDARY	0	0	No Data	0	1	0	No Data	1	Not Applicable