**Manual de Troubleshooting para AG (Always On Availability Groups) no SQL Server**

This manual serves as a guide for investigating and resolving issues in a SQL Server Always On Availability Group (AG). It covers everything from basic connectivity issues to more complex problems related to synchronization and performance.

When beginning the investigation process, I recommend using OneNote to document findings. Take screenshots of all results, errors, and key observations, ensuring that each includes a timestamp.

This approach helps create a timeline of events, provides clear evidence of the issue, and facilitates a structured analysis to determine the Root Cause Analysis (RCA) and the appropriate solution.

Contents

[**1.** **INITIAL CHECK** 4](#_Toc191581834)

[**1.1** **Checking SQL Server Version** 4](#_Toc191581835)

[**1.2** **Checking Windows Server Version** 5](#_Toc191581836)

[**1.3** **Via Command Prompt (CMD)** 6](#_Toc191581837)

[**1.4** **Summary of Commands** 6](#_Toc191581838)

[**1.5** **Using Failover Cluster Manager (GUI)** 7](#_Toc191581839)

[**2.** **Connectivity Check** 9](#_Toc191581840)

[**2.1** **Using SQL Server Configuration Manager** 9](#_Toc191581841)

[**2.2** **Checking the Always On Availability Group Endpoint Port** 10](#_Toc191581842)

[**2.3** **Testing Connectivity Between Replicas** 10](#_Toc191581843)

[**2.4** **Validation of the Always On Endpoint** 10](#_Toc191581844)

[**3.** **Checking Replica Synchronization** 11](#_Toc191581845)

[**3.1** **Checking Database Synchronization Status** 11](#_Toc191581846)

[**3.2** **Checking Transaction Log Movement** 11](#_Toc191581847)

[**3.3** **Checking for Network Latency Issues** 12](#_Toc191581848)

[I. Check Network Latency: 14](#_Toc191581849)

[II. Monitor SQL Server Error Logs: 14](#_Toc191581850)

[III. Check Disk Performance on Secondary Replica: 15](#_Toc191581851)

[**4. Failover and Cluster Investigation** 16](#_Toc191581852)

[**4.1 Check the Cluster Status** 16](#_Toc191581853)

[**4.2** **Check Quorum Status** 16](#_Toc191581854)

[**4.3** **Cluster Failure Events** 17](#_Toc191581855)

[**4.4** **Collecting Cluster Logs** 18](#_Toc191581856)

[**4.5** **Check Cluster Node Configuration** 18](#_Toc191581857)

[**5.** **Performance Check** 19](#_Toc191581858)

[**5.1** **Identify Queries Causing HADR\_SYNC\_COMMIT Waits** 19](#_Toc191581859)

[**5.2 Check Disk I/O Performance** 20](#_Toc191581860)

[**6.** **Common Troubleshooting Issues** 21](#_Toc191581861)

[**Error: Secondary Replica Not Synchronizing** 21](#_Toc191581862)

[**Error: "Data Movement is Suspended"** 21](#_Toc191581863)

[· **Error: Failover is Not Working** 22](#_Toc191581864)

[**If automatic failover is not working:** 22](#_Toc191581865)

1. **INITIAL CHECK**

Before starting the investigation, collect essential information about the environment:

* 1. **Checking SQL Server Version**

Via SQL Server Management Studio (SSMS), run the following query in SQL Server:

SELECT SERVERPROPERTY('ProductVersion') AS SQL\_Version,

SERVERPROPERTY('ProductLevel') AS Product\_Level,

SERVERPROPERTY('Edition') AS Edition,

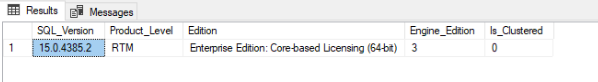
SERVERPROPERTY('EngineEdition') AS Engine\_Edition,

SERVERPROPERTY('IsClustered') AS Is\_Clustered;

✅ **What to check:**

* SQL\_Version: Indicates the SQL Server version (e.g., 15.0.2000.5 → SQL Server 2019).
* Product\_Level: Shows the product level (RTM, SP1, SP2, etc.).
* Edition: Specifies the edition (Enterprise, Standard, Express, etc.).
* Engine\_Edition: Shows the instance type (e.g., 3 = Enterprise Edition).
* Is\_Clustered: If 1, the SQL Server is part of a cluster.

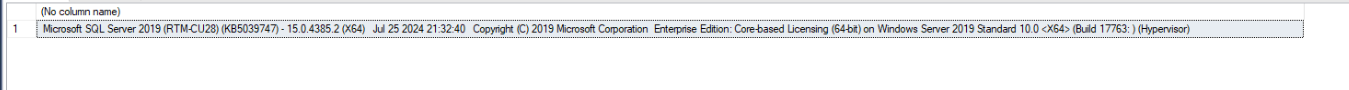
✅ **Example output:**



**Get Full SQL Server Version**

SELECT @@VERSION;

✅ **Example output:**



* 1. **Checking Windows Server Version**

Via SQL Server (Executing in SSMS)

If you want to check the Windows version directly from SQL Server, use:

EXEC xp\_cmdshell 'systeminfo';

✅ **What to check:**

* OS Name: Name of the operating system.
* OS Version: Version number (e.g., 10.0.17763 → Windows Server 2019).

⚠️ **Note:** xp\_cmdshell may be disabled by default. To enable it temporarily:

EXEC sp\_configure 'show advanced options', 1;

RECONFIGURE;

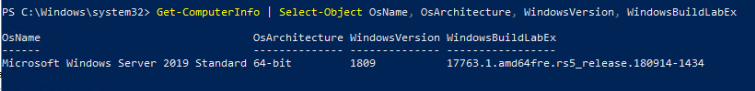
EXEC sp\_configure 'xp\_cmdshell', 1;

RECONFIGURE;

**Via PowerShell**

If accessing the server directly, run the following in PowerShell:

Get-ComputerInfo | Select-Object OsName, OsArchitecture, WindowsVersion, WindowsBuildLabEx



✅ **What to check:**

* OsName: Operating system name (e.g., **Windows Server 2019 Datacenter**).
* OsArchitecture: OS architecture (64-bit or 32-bit).
* WindowsVersion: Version number (10.0.17763 → Windows Server 2019).
* WindowsBuildLabEx: Displays build details of the system.
  1. **Via Command Prompt (CMD)**

Open the command prompt and run:

winver



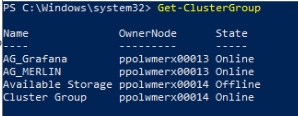
This will open a window displaying the Windows version.

* 1. **Summary of Commands**

| **Information** | **Method** | **Command** |
| --- | --- | --- |
| **SQL Server Version** | SSMS | SELECT @@VERSION; |
| **Detailed SQL Server Version** | SSMS | SELECT SERVERPROPERTY('ProductVersion'), SERVERPROPERTY('Edition'); |
| **Windows Server Version** | SSMS (cmd) | EXEC xp\_cmdshell 'systeminfo'; |
| **Windows Server Version** | PowerShell | Get-ComputerInfo |
| **Cluster WSFC Version** | PowerShell | Get-Cluster |

You can also check the primary node of the cluster using PowerShell:

Get-ClusterGroup



* 1. **Using Failover Cluster Manager (GUI)**

1. Open Failover Cluster Manager (cluadmin.msc).
2. Navigate to Roles.
3. Look for the Availability Group name.
4. The Current Owner column shows the Primary Replica node.

• AG topology (number of nodes, primary and secondary replicas, failover mode)

• Synchronization mode (Synchronous or Asynchronous)

• Synchronization status

• Recent error logs

Run the command below to view the AG status:

SELECT

ag.name AS AGName,

ar.replica\_server\_name,

ars.is\_local AS IsLocalReplica,

ars.role\_desc AS ReplicaRole,

drs.synchronization\_state\_desc,

drs.database\_state\_desc

FROM sys.availability\_groups AS ag

JOIN sys.availability\_replicas AS ar

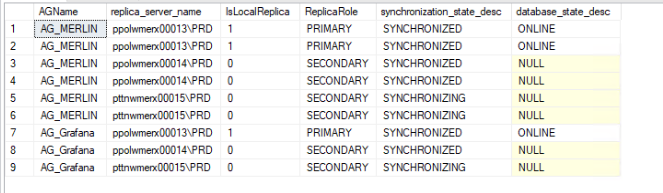
ON ag.group\_id = ar.group\_id

JOIN sys.dm\_hadr\_availability\_replica\_states AS ars

ON ar.replica\_id = ars.replica\_id

JOIN sys.dm\_hadr\_database\_replica\_states AS drs

ON ar.replica\_id = drs.replica\_id;

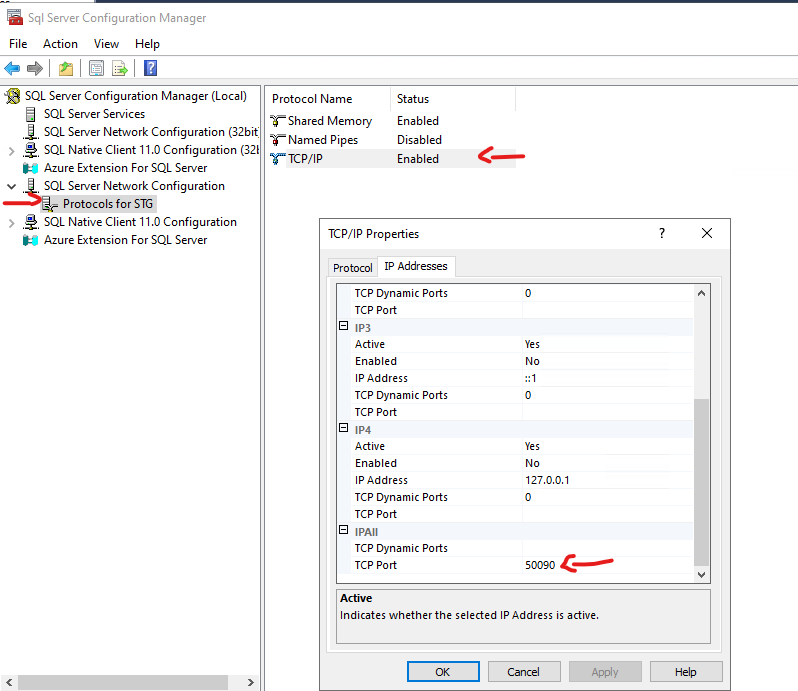


|  |  |
| --- | --- |
| Column | Description |
| AGName | Name of the Availability Group |
| replica\_server\_name | Name of the replica server |
| IsLocalReplica | 1 if the replica is local, 0 if it's a remote replica |
| ReplicaRole | Shows whether the server is PRIMARY or SECONDARY |
| synchronization\_state\_desc | Synchronization state: SYNCHRONIZED, SYNCHRONIZING, NOT SYNCHRONIZING |
| database\_state\_desc | Database state: ONLINE, RECOVERING, SUSPECT |

✅ **What to check:**

* + **synchronization\_state\_desc: If it is NOT SYNCHRONIZING, there are problems with replication.**
  + **database\_state\_desc: It should be ONLINE. If it is RECOVERING or SUSPECT, there is an error in the database.**

1. **Connectivity Check**
   1. **Using SQL Server Configuration Manager**
   2. Open SQL Server Configuration Manager.
   3. Navigate to SQL Server Network Configuration → Protocols for [YourInstanceName].
   4. Double-click TCP/IP.
   5. Under the IP Addresses tab, check the TCP Dynamic Ports and TCP Port fields:
   6. If TCP Dynamic Ports has a value (e.g., 49172), SQL Server is using a dynamic port.
   7. If TCP Port has a value (e.g., 1433), that is the static port.



* 1. **Checking the Always On Availability Group Endpoint Port**

To find the Always On Availability Group (AG) endpoint port, use:

SELECT name, protocol\_desc, type\_desc, state\_desc, port

FROM sys.tcp\_endpoints

WHERE type\_desc = 'DATABASE\_MIRRORING';



* 1. **Testing Connectivity Between Replicas**
* Check if SQL Server is listening on the correct port (powershell)

tnc ServerName -Port 1433

* Check the Always On Availability Group endpoint port (usually 5022):

tnc ServerName -Port 5022

✅ **What to check:**

If ping fails, there might be a network issue.

If TNC fails, the port may be blocked by a firewall, or the SQL Server service might not be listening properly.

If TNC fails, verify firewall rules and check if the port is listening with:

netstat -an | findstr 5022

* 1. **Validation of the Always On Endpoint**

Verify if the endpoints are correctly configured by running:

SELECT name, role\_desc, state\_desc, is\_encryption\_enabled

FROM sys.database\_mirroring\_endpoints;

A screenshot of a computerAI-generated content may be incorrect.

If the state is not STARTED, manually start the endpoint:

ALTER ENDPOINT [Hadr\_endpoint] STATE = STARTED;

1. **Checking Replica Synchronization**
   1. **Checking Database Synchronization Status**

Run the following query to check the synchronization status of the databases in the Always On Availability Group:

SELECT db\_name(database\_id) as dbname,

is\_primary\_replica,

synchronization\_state\_desc,

last\_commit\_time

FROM sys.dm\_hadr\_database\_replica\_states;

✅ **What to check:**

synchronization\_state\_desc = NOT SYNCHRONIZING → Indicates a replication failure. This could be due to network issues, database state problems, or an endpoint failure.

last\_commit\_time is significantly behind compared to the primary replica → Suggests a replication delay, which may be caused by high transaction volume, slow network, or resource constraints on the secondary replica.

* 1. **Checking Transaction Log Movement**

To identify issues with log transmission and redo queue, run:

SELECT db\_name(database\_id) as dbname,

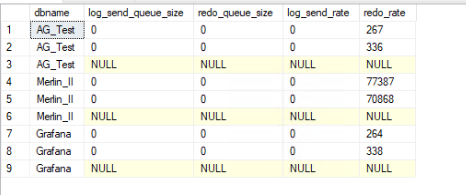
log\_send\_queue\_size,

redo\_queue\_size,

log\_send\_rate,

redo\_rate

FROM sys.dm\_hadr\_database\_replica\_states;



✅ **What to check:**

log\_send\_queue\_size (Log Send Queue Size):

If this value is high, the primary replica is generating log records faster than they can be sent to the secondary replica.

Possible causes: Network bottlenecks, blocked endpoints, or insufficient throughput on the secondary server.

redo\_queue\_size (Redo Queue Size):

If this value is high, the secondary replica is receiving logs but struggling to apply them.

This can indicate high disk latency, CPU contention, or insufficient resources on the secondary replica.

log\_send\_rate & redo\_rate:

If both are low, it may suggest network latency, storage performance issues, or insufficient system resources.

* 1. **Checking for Network Latency Issues**

To identify network latency between the primary and secondary replicas, run:

SELECT database\_id,

synchronization\_health\_desc,

synchronization\_state\_desc,

log\_send\_queue\_size,

redo\_queue\_size,

log\_send\_rate,

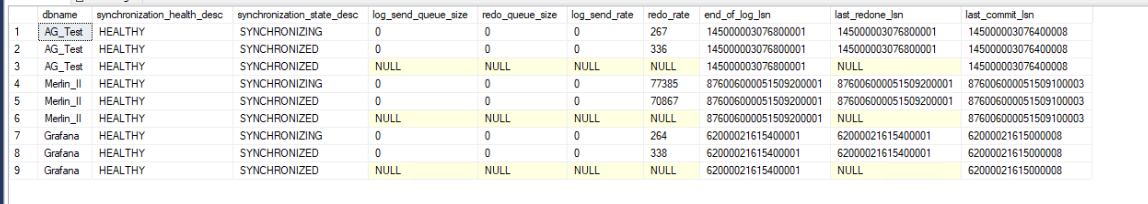
redo\_rate,

end\_of\_log\_lsn,

last\_redone\_lsn,

last\_commit\_lsn

FROM sys.dm\_hadr\_database\_replica\_states;



What to check:

synchronization\_health\_desc → Indicates the health of the replica (HEALTHY, PARTIALLY\_HEALTHY, NOT\_HEALTHY).

synchronization\_state\_desc → Current state of synchronization (SYNCHRONIZED, SYNCHRONIZING, NOT SYNCHRONIZING).

log\_send\_queue\_size → The amount of log records waiting to be sent to the secondary replica (high values indicate delays).

redo\_queue\_size → The amount of log records that have arrived at the secondary but are waiting to be applied (high values indicate that the secondary is slow in replaying logs).

log\_send\_rate → Speed at which the log records are being sent from the primary replica (low values may indicate network or performance issues).

redo\_rate → Speed at which the secondary replica is applying log records (low values may indicate disk I/O issues on the secondary replica).

end\_of\_log\_lsn → Last Log Sequence Number (LSN) recorded on the primary.

last\_redone\_lsn → Last LSN that was successfully applied (replayed) on the secondary replica.

last\_commit\_lsn → Last committed transaction log sequence number.

How to Interpret the Results

|  |  |  |
| --- | --- | --- |
| Column | Meaning | What to Check |
| synchronization\_health\_desc | Health status of the database replica | If NOT\_HEALTHY, investigate further |
| synchronization\_state\_desc | Synchronization status | If NOT SYNCHRONIZING, replication has issues |
| log\_send\_queue\_size | Amount of transaction log waiting to be sent | If high, primary may be generating logs faster than it can send |
| redo\_queue\_size | Amount of log waiting to be replayed on secondary | If high, the secondary is struggling to apply logs |
| log\_send\_rate | Speed at which logs are sent | If very low, there may be network latency issues |
| redo\_rate | Last log sequence number on the primary | If far behind end\_of\_log\_lsn, there is a replication delay |
| last\_redone\_lsn | Last log sequence number applied on the secondary | If far behind end\_of\_log\_lsn, there is a replication delay |
| last\_commit\_lsn | Last committed transaction | Helps track how up-to-date the secondary is |

If replication is slow or not synchronizing, check the following:

1. Check Network Latency:

Run a network test between replicas using PowerShell:

* the port used by the sql server, listener and endpoint must be checked

Test-NetConnection -ComputerName SecondaryReplica -Port 1433

tnc -ComputerName SecondaryReplica -Port 1433

If the connection is slow or failing, check firewall rules or network congestion.

1. Monitor SQL Server Error Logs:

Look for errors related to HADR (Always On replication issues):

EXEC xp\_readerrorlog;

Check for messages indicating log transmission failures or endpoint connectivity issues.

1. Check Disk Performance on Secondary Replica:

If redo\_rate is low but log\_send\_rate is high, the secondary may have disk I/O bottlenecks.

Use Performance Monitor (PerfMon) and check:

Avg. Disk sec/Write

Avg. Disk sec/Read

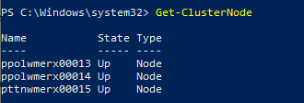
Disk Queue Length

**4. Failover and Cluster Investigation**

**4.1 Check the Cluster Status**

Run the following command to verify the state of cluster nodes:

**Get-ClusterNode**

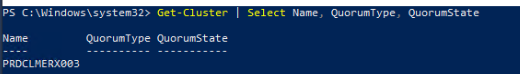
****

**✅ What to check:**

If a node is Down, there may be a communication failure or an issue with the nodeitself.

* 1. **Check Quorum Status**

**Get-Cluster | Select Name, QuorumType, QuorumState**

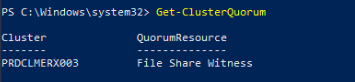
****

**✅ What to check:**

QuorumState: If the value is 0, the cluster has lost quorum and may not be operating correctly.

To check if a witness (disk, file share, or cloud witness) is set up, use:

Get-ClusterQuorum

****

* 1. **Cluster Failure Events**

To check for recent failover events in the Windows Event Log, run:

Get-WinEvent -LogName System | Where-Object {$\_.ID -eq 1135}

Get-WinEvent -LogName System | Where-Object {$\_.ID -in @(1135, 1561, 1177, 1126, 1046, 1254, 1641, 1650)}

**✅ What to check:**

Critical Cluster Failures

Event ID 1135 – A cluster node was removed due to a communication failure.

Event ID 1561 – Cluster lost quorum.

Event ID 1564 – The cluster service detected that the quorum configuration changed.

Event ID 1177 – Cluster node cannot communicate with other nodes.

Cluster Network and Communication Issues

Event ID 1126 – Cluster network name failed to register in DNS.

Event ID 1046 – Cluster service failed to bring a resource online.

Event ID 1077 – Cluster service lost connection to a node.

Failover and Resource Issues

Event ID 1254 – A node was removed from active failover membership.

Event ID 1641 – A cluster resource failed to come online.

Event ID 1650 – A failover group failed to move to another node.

* 1. **Collecting Cluster Logs**

To collect logs for a specific node over the last 5 minutes, use:

Get-ClusterLog -Node ServerName -TimeSpan 5 -Destination C:\Logs -UseLocalTime

**✅ What to check:**

-UseLocalTime ensures that the logs are in the server’s local timezone, making it easier to correlate with other events.

-TimeSpan 5 extracts logs from the last 5 minutes, useful for identifying recent failures.

To collect logs from all nodes for the last 60 minutes, run:

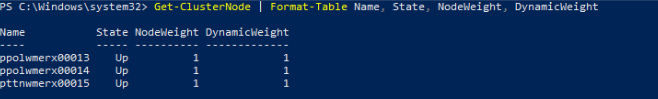
Get-ClusterLog -TimeSpan 60 -Destination C:\Logs -UseLocalTime

These logs help identify communication and quorum failures and understand recent node issues.

* 1. **Check Cluster Node Configuration**

Run the following command to verify node settings:

**Get-ClusterNode | Format-Table Name, State, NodeWeight, DynamicWeight**

****

**✅ What to check:**

NodeWeight: Should be 1 for active nodes and 0 for nodes that do not participate in quorum.

If State is Down, that node is offline and may cause issues in the cluster.

* 1. **Performance Check**
  2. **Identify Queries Causing HADR\_SYNC\_COMMIT Waits**

Queries waiting on HADR\_SYNC\_COMMIT indicate that the primary replica is waiting for the secondary to acknowledge the transaction before committing it.

SELECT r.session\_id,

r.blocking\_session\_id,

r.wait\_type,

r.wait\_time,

r.status,

r.command,

r.database\_id,

t.text AS sql\_text

FROM sys.dm\_exec\_requests AS r

JOIN sys.dm\_exec\_connections AS c

ON r.session\_id = c.session\_id

CROSS APPLY sys.dm\_exec\_sql\_text(r.sql\_handle) AS t

WHERE r.wait\_type = 'HADR\_SYNC\_COMMIT'

ORDER BY r.wait\_time DESC;

✅ What to check:

wait\_time → If high, the transaction is waiting too long for AG synchronization.

sql\_text → Identifies the query causing the delay.

blocking\_session\_id → If not NULL, another session is blocking this query, slowing replication

**5.2 Check Disk I/O Performance**

SELECT database\_id,

file\_id,

io\_stall\_read\_ms,

io\_stall\_write\_ms,

num\_of\_reads,

num\_of\_writes

FROM sys.dm\_io\_virtual\_file\_stats(NULL, NULL);

✅ What to check:

io\_stall\_read\_ms → High values indicate slow disk read performance.

io\_stall\_write\_ms → High values indicate slow disk write performance.

num\_of\_reads / num\_of\_writes → Helps assess if high I/O is expected or excessive.

* 1. **Common Troubleshooting Issues**

**Error: Secondary Replica Not Synchronizing**

Check logs in the SQL Server Error Log:

EXEC xp\_readerrorlog;

✅ **What to check:**

Look for error messages such as "Failed to send data", which indicate connectivity issues or endpoint permission problems.

**Error: "Data Movement is Suspended"**

Restart synchronization

select ag\_name, replica\_name, ag\_db\_name, synchronization\_state\_desc, 'ALTER DATABASE '+ag\_db\_name+' SET HADR RESUME;' from sys.dm\_hadr\_cached\_database\_replica\_states where synchronization\_state\_desc = 'NOT SYNCHRONIZING'

ALTER DATABASE DatabaseName SET HADR RESUME;

✅ **What to check:**

If synchronization remains suspended, check the SQL Server Error Logs for further details

* **Error: Failover is Not Working**

Verify if the failover mode is correctly set:

SELECT

ag.name AS AGName,

ar.failover\_mode\_desc

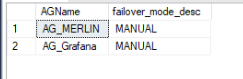
FROM sys.availability\_groups AS ag

JOIN sys.availability\_replicas AS ar

ON ag.group\_id = ar.group\_id

WHERE ar.replica\_id IN

(SELECT replica\_id FROM sys.dm\_hadr\_availability\_replica\_states WHERE is\_local = 1);



If the failover mode is incorrect, automatic failover will not occur in case of failure.

How to interpret the results:

AUTOMATIC → Failover occurs automatically if the primary replica fails (only works in Synchronous-Commit Mode).

MANUAL → Requires manual intervention to fail over to a secondary replica.

**If automatic failover is not working:**

Check the Always On Dashboard in SQL Server Management Studio (SSMS).

Verify that the failover mode is set to AUTOMATIC using the corrected query.

Ensure the secondary replica is synchronized (

SELECT \* FROM sys.dm\_hadr\_database\_replica\_states;).

Check Windows Server Failover Cluster (WSFC) logs using:

Get-ClusterLog -TimeSpan 30 -UseLocalTime

If problems persist, check for Event ID 1135 in the Windows Event Viewer, which indicates a node failure.

This manual covers the key steps for troubleshooting Always On Availability Groups in SQL Server. Using the Get-ClusterLog command is essential for obtaining detailed cluster logs and identifying issues with nodes and quorum.

If problems persist, it may be necessary to review advanced quorum configurations, optimize network settings, or even open a support case with Microsoft.