

SQL Server ALWAYSON

Interview Questions and Answers

Introduction: This chapter takes you through the SQL Server ALWAYSON related interview questions and answers. These questions are helpful for range of database administrators starting from a junior to an expert level DBA for the technical interview preparation. These questions and answers are deals with the below topics:

- SQL Server ALWAYSON
- AlwaysOn Scenarios

SQL Server AlwaysOn

1. What is ALWAYSON in SQL Server?

Ans:

ALWAYSON Availability Groups feature is a high-availability and disaster-recovery solution that provides an enterprise-level alternative to database mirroring. Introduced in SQL Server 2012, ALWAYSON Availability Groups maximizes the availability of a set of user databases for an enterprise. An availability group supports a failover environment for a discrete set of user databases, known as *availability databases* that fail over together. An availability group supports a set of read-write primary databases and one to four sets of corresponding secondary databases. Optionally, secondary databases can be made available for read-only access and/or some backup operations.

2. What are Availability Groups?

Ans:

A container for a set of databases, availability databases, that fails over together. Let's consider a scenario where a set of 3 databases are interlinked based on application requirement. Now we need to setup HA for these 3 databases. If we choose mirroring we need to have a separate mirroring setup for these 3 databases whereas in ALWAYSON Availability Groups easier the job by grouping all these 3 databases.

3. What are Availability Databases?

Ans:

It's a database that belongs to an availability group. For each availability database, the availability group maintains a single read-write copy (the primary database) and one to four read-only copies (secondary databases).

4. Which SQL/Windows Server Editions include ALWAYSON Availability Group functionality?

Ans:

SQL Server Enterprise Edition and Windows Enterprise Edition

5. How many replicas can I have in an ALWAYSON Availability Group?

Ans:

SQL Server 2012: 1 Primary and up to 4 Secondary's

SQL Server 2014: 1 Primary and up to 8 Secondary's

6. What are the ALWAYSON improvements added in SQL Server 2014?

Ans:

Enhanced Availability for Read-Only Replicas:

With SQL Server 2014, the availability of secondary replicas

has been enhanced to allow read workloads to continue to run even in the case of lengthy network failures or the loss of quorum for the Windows Server Failover Cluster.

Increased Number of Replicas:

SQL Server 2012 supported a maximum of four secondary replicas. With SQL Server 2014, ALWAYSON Availability Groups now supports up to eight secondary replicas.

Integration with Windows Azure:

On-premises SQL Server instances can use the new Windows Azure configuration options in the ALWAYSON Availability Group wizard to create one or more asynchronous secondary replicas on Windows Azure Infrastructure as a Service (IaaS) services.

SQL Server 2014 ALWAYSON Availability Groups can provide high availability for SQL Server databases hosted in Windows

Azure.

Enhanced Diagnostics:

SQL Server 2014 ALWAYSON Availability Groups diagnostic and troubleshooting message display has been improved to offer more specific information. Additional columns have also been made more discoverable in the ALWAYSON Dashboard.

7. What are the ALWAYSON improvements in SQL Server 2016?

Ans:

Round-robin load balancing in readable SECONDARIES:

In SQL Server 2014 redirecting activity to the secondary replicas happens through read-only routing list. But the first replica in the list gets the most activity because it's the one that is always tried first. In SQL Server 2016, the list of readable secondary replicas offers up connection information on a

round robin basis. Also, each replica has its own read-only routing list so that read-only balancing via the availability group listener could route traffic to secondary replicas.

Increased number of auto-failover targets:

In SQL Server 2014, there can be three synchronous replicas, but only two of those can be designated as automatic failover targets. In SQL Server 2016, all three synchronous replicas can now be designated as failover targets.

Distributed Transaction Coordinator support:

SQL Server 2016 ALWAYSON Availability Groups supports the Distributed Transaction Coordinator (DTC) to manage transactions across multiple databases and instances. Remember to get this support SQL Server should be in Windows Server 2016 or 2012 R2 with the latest rollup patch.

Failover on database health:

In 2014 and 2012 failover is depends on SQL Server instance health but starting from 2016 failover can be based on database health. Ex: One of the databases is offline in an instance:

- **2014/2012:** Since instance and other databases are running fine it doesn't trigger a failover
- **2016:** We can control the failover behavior based on database health; in this case all databases within the affected AG would failover.

Enhanced log replication throughput and redo speed:

Microsoft has worked to streamline the pipeline between the synchronous replicas to gain better log-data throughput when utilizing SQL Server ALWAYSON Availability Groups.

Support for group-managed service accounts:

A group Managed Service Account (gMSA) is a type of security account released in Windows Server 2012 and improved in SQL Server 2016. If the instance is part of an AG, setting up and maintaining permissions to access common resources such as file shares often becomes complex as security provisions need to be established for the service account from each instance within the AG. The group account provides similar security capabilities as a local managed service account on an individual server, but it has a domain scope. gMSA can automatically manage password synchronization across all nodes.

8. How many ALWAYSON Availability Groups can be configured in Always ON?

Ans:

Up to 10 availability groups is the recommendation, but it's not

enforced

9. How many databases can be configured in an ALWAYSON Availability Group?

Ans:

Up to 100 is the recommendation, but it's not enforced

10. What are the Restrictions on Availability Groups?

Ans:

- Availability replicas must be hosted by different nodes of one WSFC cluster
- Unique availability group name: Each availability group name must be unique on the WSFC cluster. The maximum length for an availability group name is 128 characters.
- Availability replicas: Each availability group supports one primary replica and up to four secondary replicas. All of the

replicas can run under asynchronous-commit mode, or up to three of them can run under synchronous-commit mode.

- Maximum number of availability groups and availability databases per computer: The actual number of databases and availability groups you can put on a computer (VM or physical) depends on the hardware and workload, but there is no enforced limit. Microsoft has extensively tested with 10 AGs and 100 DBs per physical machine.
- Do not use the Failover Cluster Manager to manipulate availability groups.

11. What are the minimum requirements of a database to be part of the Always ON Availability Group?

Ans:

- Availability groups must be created with user databases. Systems databases can't be used.
- Databases must be read-write. Read-only databases aren't supported.
- Databases must be multiuser databases.
- Databases can't use the AUTO_CLOSE feature.
- Databases must use the full recovery model, and there must be a full backup available.
- A given database can only be in a single availability group, and that database can't be configured to use database mirroring.

12. How many read-write and read only databases replica can be configure in SQL Server 2012 and 2014?

Ans:

- SQL Server 2012 supported a maximum of four secondary replicas.
- With SQL Server 2014, ALWAYSON Availability Groups now supports up to eight secondary replicas.

13. Is it possible to setup Log Shipping on a database which is part of Availability Group?

Ans:

Yes, it can be configured.

14. Is it possible to setup Replication on a database which is part of Availability Group?

Ans:

Yes, it is possible.

15. Does FILESTEAM, Change Data Capture and Database Snapshot supported by Availability Group?

Ans:

Yes, all these features are supported by ALWAYSON Availability Group.

16. Can system database participate in AG?

Ans:

No.

17. What version of Windows do I need for ALWAYSON AGs?

Ans:

We highly recommend Windows Server 2012R2 and above.

18. Can I have different indexes or tables on my replicas?

Ans:

No, the replica database contents will be exactly the same as the primary.

19. What is Availability mode in Always ON?

Ans:

The availability mode is a property of each availability replica. The availability mode determines whether the primary replica waits to commit transactions on a database until a given secondary replica has written the transaction log records to disk (hardened the log). ALWAYSON supports below modes:

Asynchronous-commit mode: Primary replica commits the transaction on a database without waiting for the conformation from the secondary replica.

Synchronous-commit mode: Primary replica does not commit the transaction on a database until it gets the confirmation (written the transaction log records to disk on secondary) from secondary replica.

20. What is the Difference between Asynchronous-commit

mode and Synchronous-commit mode?

Ans:

Asynchronous-commit mode:

An availability replica that uses this availability mode is known as an asynchronous-commit replica. Under asynchronous-commit mode, the primary replica commits transactions without waiting for acknowledgement that an asynchronous-commit secondary replica has hardened the log. Asynchronous-commit mode minimizes transaction latency on the secondary databases but allows them to lag behind the primary databases, making some data loss possible.

Synchronous-commit mode:

An availability replica that uses this availability mode is known as a synchronous-commit replica. Under synchronous-commit mode, before committing transactions, a synchronous-commit

primary replica waits for a synchronous-commit secondary replica to acknowledge that it has finished hardening the log. Synchronous-commit mode ensures that once a given secondary database is synchronized with the primary database, committed transactions are fully protected. This protection comes at the cost of increased transaction latency.

21. What is called Primary replica?

Ans:

The availability replica that makes the primary databases available for read-write connections from clients is called Primary Replica. It sends transaction log records for each primary database to every secondary replica.

22. What is called Secondary replica?

Ans:

An availability replica that maintains a secondary copy of each

availability database, and serves as a potential failover targets for the availability group. Optionally, a secondary replica can support read-only access to secondary databases can support creating backups on secondary databases.

23. What is Availability Group listener?

Ans:

Availability Group Listener is a server name to which clients can connect in order to access a database in a primary or secondary replica of an ALWAYSON availability group. Availability group listeners direct incoming connections to the primary replica or to a read-only secondary replica.

24. What are Readable Secondary Replicas?

Ans:

The ALWAYSON Availability Groups active secondary capabilities include support for read-only access to one or more

secondary replicas (readable secondary replicas). A readable secondary replica allows read-only access to all its secondary databases. However, readable secondary databases are not set to read-only. They are dynamic. A given secondary database changes as changes on the corresponding primary database are applied to the secondary database.

25. What are the benefits of Readable Secondary Replicas?

Ans:

Directing read-only connections to readable secondary replicas provides the following benefits:

- Offloads your secondary read-only workloads from your primary replica, which conserves its resources for your mission critical workloads. If you have mission critical read-workload or the workload that cannot tolerate latency, you should run it on the primary.

- Improves your return on investment for the systems that host readable secondary replicas.
- In addition, readable SECONDARIES provide robust support for read-only operations, as follows:
 - ✓ Temporary statistics on readable secondary database optimize read-only queries. For more information, see Statistics for Read-Only Access Databases, later in this topic.
 - ✓ Read-only workloads use row versioning to remove blocking contention on the secondary databases. All queries that run against the secondary databases are automatically mapped to snapshot isolation transaction level, even when other transaction isolation levels are explicitly set. Also, all locking hints are ignored. This eliminates reader/writer

contention.

26. How many synchronous secondary replicas can I have?

Ans:

We can have up to 2 synchronous replicas, but we are not required to use any. We could run all SECONDARIES in asynchronous mode if desired

27. Can we use a secondary for reporting purpose?

Ans:

Yes. An active secondary can be used to offload read-only queries from the primary to a secondary instance in the availability group.

28. Can we use secondary replicas to take the DB backups?

Ans:

Yes. An active secondary can be used for some types of backups

29. What all types of DB backups are possible on Secondary Replicas?

Ans:

- **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.

30. What is “Failover” in ALWAYSON?

Ans:

Within the context of a session between the primary replica and a secondary replica, the primary and secondary roles are

potentially interchangeable in a process known as failover. During a failover the target secondary replica transitions to the primary role, becoming the new primary replica. The new primary replica brings its databases online as the primary databases, and client applications can connect to them. When the former primary replica is available, it transitions to the secondary role, becoming a secondary replica. The former primary databases become secondary databases and data synchronization resumes.

31. How many types of Failover are supported by Always ON?

Ans:

Three forms of failover exist—automatic, manual, and forced (with possible data loss). The form or forms of failover supported by a given secondary replica depends on its

availability mode.

32. What are the Failover types supported by Synchronous-commit mode?

Ans:

- **Planned manual failover** (without data loss)
- **Automatic failover** (without data loss)

33. What is planned manual failover?

Ans:

A manual failover occurs after a database administrator issues a failover command and causes a synchronized secondary replica to transition to the primary role (with guaranteed data protection) and the primary replica to transition to the secondary role. A manual failover requires that both the primary replica and the target secondary replica are running

under synchronous-commit mode, and the secondary replica must already be synchronized.

34. What is Automatic failover?

Ans:

An automatic failover occurs in response to a failure that causes a synchronized secondary replica to transition to the primary role (with guaranteed data protection). When the former primary replica becomes available, it transitions to the secondary role. Automatic failover requires that both the primary replica and the target secondary replica are running under synchronous-commit mode with the failover mode set to "Automatic." In addition, the secondary replica must already be synchronized, have WSFC quorum, and meet the conditions specified by the flexible failover policy of the availability group.

35. What are the Failover types supported by under

asynchronous-commit mode?

Ans:

Only form of failover is forced manual failover (with possible data loss), typically called **forced failover**. **Forced failover** is considered a form of manual failover because it can only be initiated manually. Forced failover is a disaster recovery option. It is the only form of failover that is possible when the target secondary replica is not synchronized with the primary replica.

36. What is the use of ALWAYSON Dashboard?

Ans:

Database administrators use the ALWAYSON Dashboard to obtain an at-a-glance view of the health of an ALWAYSON availability group and its availability replicas and databases in SQL Server 2012. Some of the typical uses for the ALWAYSON

Dashboard are:

- Choosing a replica for a manual failover.
- Estimating data loss if you force failover.
- Evaluating data-synchronization performance.
- Evaluating the performance impact of a synchronous-commit secondary replica

37. What is availability group wizard?

Ans:

Availability Group Wizard is a GUI using SQL Server Management Studio to create and configure an ALWAYSON availability group in SQL Server 2012.

38. Suppose primary database became in suspect mode. Will AG have failover to secondary replica?

Ans:

Issues at the database level, such as a database becoming suspect due to the loss of a data file, deletion of a database, or corruption of a transaction log, do not cause an availability group to failover.

39. Can we have two primary availability replicas?

Ans:

No, it is not possible.

40. Does AG support automatic page repair for protection against any page corruption happens?

Ans:

Yes, it automatically takes care of the automatic page repair.

41. How to add a secondary database from an availability group using T-SQL?

Ans:

ALTER DATABASE Db1 SET HADR AVAILABILITY GROUP =

<AGName>;

42. How to remove a secondary database from an availability group?

Ans:

ALTER DATABASE <DBName> SET HADR OFF;

43. SQL Server 2012 ALWAYSON supports encryption and compression?

Ans:

SQL Server 2012 ALWAYSON Availability Group supports row and page compression for tables and indexes, we can use the data compression feature to help compress the data inside a database, and to help reduce the size of the database. We can use encryption in SQL Server for connections, data, and stored procedures; we can also perform database level encryption: Transparent data encryption (TDE). If you use transparent data

encryption (TDE), the service master key for creating and decrypting other keys must be the same on every server instance that hosts an availability replica for the availability group

44. Does AG support Bulk-Logged recovery model?

Ans:

No, it does not.

45. Can a database belong to more than one availability group?

Ans:

No. It's not allowed.

46. What is session timeout period?

Ans:

Session-timeout period is a replica property that controls how many seconds (in seconds) that an availability replica waits for

a ping response from a connected replica before considering the connection to have failed. By default, a replica waits 10 seconds for a ping response. This replica property applies only the connection between a given secondary replica and the primary replica of the availability group.

47. How to change the Session Timeout period?

Ans:

```
ALTER AVAILABILITY GROUP <AG Name>
    MODIFY REPLICA ON '<Instance Name>' WITH
        (SESSION_TIMEOUT = 15);
```

48. What are different synchronization preferences are available?

Ans:

As part of the availability group creation process, we have to make an exact copy of the data on the primary replica on the

secondary replica. This is known as the initial data synchronization for the Availability Group.

49. How many types of Data synchronization preference options are available in Always ON?

Ans:

There are three options- Full, Join only, or Skip initial data synchronization.

50. Is it possible to run DBCC CHECKDB on secondary replicas?

Ans:

Yes.

51. Can I redirect the read-only connections to the secondary replica instead of Primary replica?

Ans:

Yes, we can specify the read_only intent in the connection

string and add only SECONDARIES (not the primary) to the read_only_routing list. If you want to disallow direct connections to the primary from read_only connections, then set its allow_connections to read_write.

52. If a DBA expands a data file manually on the primary; will SQL Server automatically grow the same file on SECONDARIES?

Ans:

Yes! It will be automatically expanded on the Secondary replica.

53. Is it possible to create additional indexes on read-only secondary replicas to improve query performance?

Ans:

No, it is not possible.

54. Is it possible to create additional statistics on read-only

SECONDARIES to improve query performance?

Ans:

No. But we can allow SQL Server to automatically create statistics on read-only secondary replicas.

55. Can we manually fail over to a secondary replica?

Ans:

Yes. If the secondary is in synchronous-commit mode and is set to “SYNCHRONIZED” you can manually fail over without data loss. If the secondary is not in a synchronized state then a manual failover is allowed but with possible data loss

56. What is read intent option?

Ans:

There are two options to configure secondary replica for running read workload. The first option ‘Read-intent-only’ is used to provide a directive to ALWAYSON secondary replica to

accept connections that have the property ApplicationIntent = ReadOnly set. The word ‘intent’ is important here as there is no application check made to guarantee that there are no DDL/DML operations in the application connecting with ‘ReadOnly’ but an assumption is made that customer will only connect read workloads.

57. Does ALWAYSON Availability Groups repair the data page corruption as Database Mirroring?

Ans:

Yes. If a corrupt page is detected, SQL Server will attempt to repair the page by getting it from another replica.

58. What are the benefits of Always on feature?

Ans:

- Utilizing database mirroring for the data transfer over

TCP/IP

- providing a combination of Synchronous and Asynchronous mirroring
- providing a logical grouping of similar databases via Availability Groups
- Creating up to four readable secondary replicas
- Allowing backups to be undertaken on a secondary replica
- Performing DBCC statements against a secondary replica
- Employing Built-in Compression & Encryption

59. How much network bandwidth is required for a successful AlwaysOn AG setup?

Ans:

For a really rough estimate, sum up the amount of uncompressed transaction log backups that you generate in a 24-hour period. You'll need to push that amount of data per day across the wire. Things get trickier when you have multiple replicas – the primary pushes changes out to all replicas, so if you've got 3 replicas in your DR site, you'll need 3x the network throughput. Calculating burst requirements is much more difficult – but at least this helps you get started.

60. What's the performance overhead of a synchronous replica?

Ans:

From the primary replica, ping the secondary, and see how long (in milliseconds) the response takes. Then run load tests on the secondary's transaction log drive and see how long writes take. That's the minimum additional time that will be added to each

transaction on the primary. To reduce the impact, make sure your network is low-latency and your transaction log drive writes are fast.

61. How far behind will my asynchronous replica be?

Ans:

The faster your network and your servers are, and the less transactional activity you have, the more up-to-date each replica will be. I've seen setups where the replicas are indistinguishable from the primary. However, I've also seen cases with underpowered replicas, slow wide area network connections, and heavy log activity (like index maintenance) where the replicas were several minutes behind.

62. What's the difference between AGs in SQL 2012 and SQL 2014?

Ans:

SQL Server 2014's biggest improvement is that the replica's databases stay visible when the primary drops offline – as long as the underlying cluster is still up and running. If I have one primary and four secondary replicas, and I lose just my primary, the SECONDARIES are still online servicing read-only queries. (Now, you may have difficulties connecting to them unless you're using the secondary's name, but that's another story.) Back in SQL 2012, when the primary dropped offline, all of the SECONDARIES' copies immediately dropped offline – breaking all read-only reporting queries.

63. How do I monitor ALWAYSON Availability Groups?

Ans:

That's rather challenging right now. Uptime monitoring means knowing if the listener is accepting writeable connections, if it's correctly routing read-only requests to other servers, if all

read-only replicas are up and running, if load is distributed between replicas the way you want, and how far each replica is running behind. Performance monitoring is even tougher – each replica has its own statistics and execution plans, so queries can run at totally different speeds on identical replicas.

AlwaysOn Scenarios

64. Can we configure Automatic failover of Availability Groups with SQL Server Failover cluster instances?

Ans:

SQL Server Failover Cluster Instances (FCIs) do not support automatic failover by availability groups, so any availability replica that is hosted by an FCI can only be configured for manual failover.

65. Can we take Transaction log backups on the secondary

replicas?

Ans:

Yes, we can take transaction log backups on the secondary replicas without COPY_ONLY option.

66. Do we need SQL Server Cluster instances to configure Always ON?

Ans:

No we don't need SQL Server Cluster instances to configure Always ON.

67. Do we need shared storage to configure Always ON?

Ans:

No, we don't need shared storage.

68. How does licensing work with ALWAYSON Availability Groups in SQL 2012 and 2014?

Ans:

All replicas have to have Enterprise Edition. If you run queries, backups, or DBCCs on a replica, you have to license it. For every server licensed with Software Assurance, you get one standby replica for free – but only as long as it's truly standby, and you're not doing queries, backups, or DBCCs on it.

69. Can I use ALWAYSON Availability Groups with Standard Edition?

Ans:

Not at this time, but it's certainly something folks have been asking for since database mirroring has been deprecated.

70. Do ALWAYSON AGs require shared storage or a SAN?

Ans:

No, you can use local storage, like cheap SSDs.

71. Do Availability Groups require a Windows cluster?

Ans:

Yes, they're built atop Windows failover clustering. This is the same Windows feature that also enables failover clustered instances of SQL Server, but you don't have to run a failover clustered instance in order to use ALWAYSON Availability Groups.

72. Do I need a shared quorum disk for my cluster?

Ans:

No

73. If I fail over to an asynchronous replica, and its behind, how do I sync up changes after the original primary comes back online?

Ans:

When I go through an AG design with a team, we talk about the work required to merge the two databases together. If it's complex (like lots of parent/child tables with identity fields,

and no update date stamp field on the tables), then management agrees to a certain amount of data loss upon failover. For example, “If we’re under fifteen minutes of data is involved, we’re just going to walk away from it.” Then we build a project plan for what it would take to actually recover >15 minutes of data, and management decides whether they want to build that tool ahead of time, or wait until disaster strikes.

74. We have got an alert “WSFC cluster service is offline.”
What is your action plan?

Ans:

This alert is raised when the WSFC cluster is offline or in the forced quorum state. All availability groups hosted within this cluster are offline (a disaster recovery action is required).

Possible Reasons:

This issue can be caused by a cluster service issue or by the loss

of the quorum in the cluster.

Possible Solutions:

Use the Cluster Administrator tool to perform the forced quorum or disaster recovery workflow. Once WFSC is started you must re-evaluate and reconfigure NodeWeight values to correctly construct a new quorum before bringing other nodes back online. Otherwise, the cluster may go back offline again.

Reestablishment may require if there are any High Availability features (ALWAYSON Availability Groups, Log Shipping, Database Mirroring) using on effected nodes.

75. How to force a WSFC (Windows Server Failover Cluster) Cluster to start without a quorum?

Ans:

This can be done using

- Failover Cluster Manager
- Net.exe
- PowerShell

Here we'll see how this can be done using FCM (Failover Cluster Manager)

- Open a Failover Cluster Manager and connect to the desired cluster node to force online.
- In the Actions pane, click Force Cluster Start, and then click Yes – Force my cluster to start.
- In the left pane, in the Failover Cluster Manager tree, click the cluster name.
- In the summary pane, confirm that the current Quorum Configuration value is: Warning: Cluster is running in Force Quorum state.

Here we'll see how this can be done using FCM (Failover Cluster Manager)

- Open a Failover Cluster Manager and connect to the desired cluster node to force online.
- In the Actions pane, click Force Cluster Start, and then click Yes – Force my cluster to start.
- In the left pane, in the Failover Cluster Manager tree, click the cluster name.
- In the summary pane, confirm that the current Quorum Configuration value is: Warning: Cluster is running in Force Quorum state.

76. We have got an alert “Availability group is offline.” Can you explain about this warning and your action plan?

Ans:

This alert is raised when the cluster resource of the availability group is offline or the availability group does not have a primary replica.

Possible Reasons:

- The availability group is not configured with automatic failover mode. The primary replica becomes unavailable and the role of all replicas in the availability group become RESOLVING.
- The availability group is configured with automatic failover mode and does not complete successfully.
- The availability group resource in the cluster becomes offline.
- There is an automatic, manual, or forced failover in progress for the availability group.

Possible Solutions:

- If the SQL Server instance of the primary replica is down, restart the server and then verify that the availability group recovers to a healthy state.
- If the automatic failover appears to have failed, verify that the databases on the replica are synchronized with the previously known primary replica, and then failover to the primary replica. If the databases are not synchronized, select a replica with a minimum loss of data, and then recover to failover mode.
- If the resource in the cluster is offline while the instances of SQL Server appear to be healthy, use Failover Cluster Manager to check the cluster health or other cluster issues on the server. You can also use the Failover Cluster Manager to attempt to turn the availability group resource online.

- If there is a failover in progress, wait for the failover to complete.
77. We have got an alert “Availability group is not ready for automatic failover.” Can you explain about this warning and your action plan?

Ans:

This alert is raised when the failover mode of the primary replica is automatic; however none of the secondary replicas in the availability group are failover ready.

Possible Reasons:

The primary replica is configured for automatic failover; however, the secondary replica is not ready for automatic failover as it might be unavailable or its data synchronization state is currently not SYNCHRONIZED.

Possible Solutions:

- Verify that at least one secondary replica is configured as automatic failover. If there is not a secondary replica configured as automatic failover, update the configuration of a secondary replica to be the automatic failover target with synchronous commit.
 - Use the policy to verify that the data is in a synchronization state and the automatic failover target is SYNCHRONIZED, and then resolve the issue at the availability replica.
78. In your environment data inserted on Primary replica but not able to see that on secondary replica. When you check that Availability is in healthy state and in most cases data reflects in a few minutes but in this case it's didn't happen. Now you need to check for the bottleneck and fix

the issue. Can you explain your views and workaround in this situation?

Ans:

Possible Reasons:

- Long-Running Active Transactions
- High Network Latency or Low Network Throughput Causes Log Build-up on the Primary Replica
- Another Reporting Workload Blocks the Redo Thread from Running
- Redo Thread Falls behind Due to Resource Contention

Possible Workaround:

- Use DBCC OPENTRAN and check if there are any oldest transactions running on primary replica and see if they can

be rolled back.

- A high DMV (`sys.dm_hadr_database_replica_states`) value `log_send_queue_size` can indicate logs being held back at the primary replica. Dividing this value by `log_send_rate` can give you a rough estimate on how soon data can be caught up on the secondary replica.
- Check two performance objects `SQL Server:Availability Replica > Flow Control Time (ms/sec)` and `SQL Server:Availability Replica > Flow control/sec`. Multiplying these two values shows you in the last second how much time was spent waiting for flow control to clear. The longer the flow control wait time, the lower the send rate.
- When the redo thread is blocked, an extended event called `sqlserver.lock_redo_blocked` is generated. Additionally, you can query the DMV `sys.dm_exec_request` on the secondary

replica to find out which session is blocking the REDO thread, and then you can take corrective action. You can let the reporting workload to finish, at which point the redo thread is unblocked. You can unblock the redo thread immediately by executing the KILL command on the blocking session ID. The following query returns the session ID of the reporting workload that is blocking the redo thread.

Transact-SQL

```
Select    session_id,    command,    blocking_session_id,  
wait_time, wait_type, wait_resource  
from sys.dm_exec_requests  
where command = 'DB STARTUP'
```

- When Redo Thread Falls Behind Due to Resource Contention; a large reporting workload on the secondary

replica has slowed down the performance of the secondary replica, and the redo thread has fallen behind. You can use the following DMV query to see how far the redo thread has fallen behind, by measuring the difference between the gap between last_redone_lsn and last_received_lsn.

Transact-SQL

```
Select recovery_lsn, truncation_lsn, last_hardened_lsn,  
last_received_lsn, last_redone_lsn, last_redone_time  
from sys.dm_hadr_database_replica_states.
```

If you see thread is indeed failing behind, do a proper investigation and take the help of resource governor and can control the CPU cycles

Note: Have a look at MSDN sites and try to understand these solutions because when you say possible solutions,

immediately you might be asked about resolutions.

79. You perform a forced manual failover on an availability group to an asynchronous-commit secondary replica; you find that data loss is more than your recovery point objective (RPO). Or, when you calculate the potential data loss of an asynchronous-commit secondary replica using the method in Monitor Performance for ALWAYSON Availability Groups, you find that it exceeds your RPO. What are the possible reasons that causes data loss is more than your recovery point objective?

Ans:

There are mainly two reasons:

- **High Network Latency or Low Network Throughput Causes Log Build-up on the Primary Replica.** The primary

replica activates flow control on the log send when it has exceeded the maximum allowable number of unacknowledged messages sent over to the secondary replica. Until some of these messages have been acknowledged, no more log blocks can be sent to the secondary replica. Since data loss can be prevented only when they have been hardened on the secondary replica, the build-up of unsent log messages increases potential data loss.

- **Disk I/O Bottleneck Slows Down Log Hardening on the Secondary Replica.** If the log file and the data file are both mapped to the same hard disk, reporting workload with intensive reads on the data file will consume the same I/O resources needed by the log hardening operation. Slow log hardening can translate to slow acknowledgement to the

primary replica, which can cause excessive activation of the flow control and long flow control wait times.

80. After an automatic failover or a planned manual failover without data loss on an availability group, you find that the failover time exceeds your recovery time objective (RTO). Or, when you estimate the failover time of a synchronous-commit secondary replica (such as an automatic failover partner) using the method in Monitor Performance for ALWAYSON Availability Groups, you find that it exceeds your RTO. Can you explain the possible reasons which causes the failover time exceeds defined RTO?

Ans:

- *Reporting Workload Blocks the Redo Thread from Running:*

On the secondary replica, the read-only queries acquire schema stability (Sch-S) locks. These Sch-S locks can block the redo thread from acquiring schema modification (Sch-M) locks to make any DDL changes. A blocked redo thread cannot apply log records until it is unblocked. Once unblocked, it can continue to catch up to the end of log and allow the subsequent undo and failover process to proceed.

- ***Redo Thread Falls Behind Due to Resource Contention:*** When applying log records on the secondary replica, the redo thread reads the log records from the log disk, and then for each log record it accesses the data pages to apply the log record. The page access can be I/O bound (accessing the physical disk) if the page is not already in the buffer pool. If there is I/O bound reporting workload, the reporting workload competes for I/O resources with the redo thread

and can slow down the redo thread.

81. Let's say you have configured Automatic failover on SQL Server 2012 ALWAYSON environment. An automatic failover triggered but unsuccessful in making secondary replica as PRIMARY. How do you identify that failover is not successful and what are the possible reasons that causes an unsuccessful failover?

Ans:

If an automatic failover event is not successful, the secondary replica does not successfully transition to the primary role. Therefore, the availability replica will report that this replica is in Resolving status. Additionally, the availability databases report that they are in Not Synchronizing status, and applications cannot access these databases.

Possible Reasons for Unsuccessful Failover:

- “**Maximum Failures in the Specified Period**” value is exhausted: The availability group has Windows cluster resource properties, such as the Maximum Failures in the Specified Period property. This property is used to avoid the indefinite movement of a clustered resource when multiple node failures occur.
- **Insufficient NT Authority\SYSTEM account permissions:** The SQL Server Database Engine resource DLL connects to the instance of SQL Server that is hosting the primary replica by using ODBC in order to monitor health. The logon credentials that are used for this connection are the local SQL Server NT AUTHORITY\SYSTEM login account. By default, this local login account is granted the following permissions: 1.Alter Any Availability Group, 2.Connect SQL, 3.View server state. If the NT AUTHORITY\SYSTEM

login account lacks any of these permissions on the automatic failover partner (the secondary replica), then SQL Server cannot start health detection when an automatic failover occurs. Therefore, the secondary replica cannot transition to the primary role. To investigate and diagnose whether this is the cause, review the Windows cluster log.

- ***The availability databases are not in a SYNCHRONIZED state:*** In order to automatically fail over, all availability databases that are defined in the availability group must be in a SYNCHRONIZED state between the primary replica and the secondary replica. When an automatic failover occurs, this synchronization condition must be met in order to make sure that there is no data loss. Therefore, if one availability database in the availability group in the

synchronizing or not synchronized state, automatic failover will not successfully transition the secondary replica into the primary role.

82. Have you ever seen the Error 41009?

Ans:

Yes! This error might occur when you try to create multiple availability groups in a SQL Server 2012 ALWAYSON failover clustering environment. This issue can be resolved by applying Cumulative Update Package 2.

83. Let's say you added a new file to a database which is a part of ALWAYSON Availability Groups. The add file operation succeeded on primary replica but failed in secondary replica. What is the impact and how you troubleshoot?

Ans:

This might happen due to a different file path between the systems that hosts primary and secondary replica. Failed add-file operation will cause the secondary database to be suspended. This, in turn, causes the secondary replica to enter the NOT SYNCHRONIZING state.

Resolution:

- Remove the secondary database from the availability group.
- On the existing secondary database, restore a full backup of the FILEGROUP that contains the added file to the secondary database, using WITH NORECOVERY and WITH MOVE (Specify the correct file path as per secondary).
- Back up the transaction log that contains the add-file

operation on the primary database, and manually restore the log backup on the secondary database using WITH NORECOVERY and WITH MOVE. Restore the last transaction log file with NO RECOVERY.

- Rejoin the secondary database to the availability group.

84. Can you write T-SQL statement for joining a replica to availability group? (AG name “ProAG”

Ans:

Connect to the server instance that hosts the secondary replica and issue the below statement:

```
ALTER AVAILABILITY GROUP ProAG JOIN;
```

The same operation can be done using SSMS or using Power Shell

85. Data synchronization state for one of the availability

database is not healthy. Can you tell me the possible reasons?

Ans:

If this is an asynchronous-commit availability replica, all availability databases should be in the SYNCHRONIZING state.

If this is a synchronous-commit availability replica, all availability databases should be in the SYNCHRONIZED state.

This issue can be caused by the following:

- The availability replica might be disconnected.
- The data movement might be suspended.
- The database might not be accessible.
- There might be a temporary delay issue due to network latency or the load on the primary or secondary replica.

86. Let's say we have a premium production server and it is

in ALWAYSON Availability Group. You observe that CPU utilization is hitting top at a specific time in a day. You did an RCA and found that CPU utilization reaches top and most CPU is from backup process due to backup compression is on. Now what do you suggest? Do we have any features for backup?

Ans:

Yes! There is an option to perform backup from secondary replicas. We can set this from Availability Group properties we can find “Backup Preferences” and from that we can choose one of the options from:

Preferred Secondary: Backups performed on Secondary if there is no secondary configured performed from primary

Secondary Only: Backups should be done from secondary only

Primary: Must occur on Primary Replica

Any Replica: Can occur from any replica in Availability Group

87. Is there any specific limitations if we need to perform auto backups from secondary backups?

Ans:

Yes! There are few:

- Only Copy_Only backup allowed from secondary replica
- Differential backups not allowed from secondary replica.
- Log backups can be performed from different secondary replicas but all these backups maintains a single log chain (LSN sequence). It might help in some of the situations

88. Have you ever applied patches/CU/service packs on ALWAYSON Availability Groups? Did you face any issues while applying?

Ans:

Yes! I have applied CU and service packs on SQL Server 2012 SP2 Cumulative Update 4

I had a bad experience with ALWAYSON AG:

After CU4 applied we saw that ALWAYSON Availability Groups are in Non-Synchronizing state.

After RCA we found that there was a huge blocking between user sessions and a unknown session, CHECKPOINT with command running as “DB_STARTUP.”

Through of the MSDN SITE we found that Microsoft declared it's a bug and the solution chosen as below:

- We had to open an outage:
- Disable Automatic Failover
- Restart the SQL Server on Primary Replica
- Re-enable automatic failover.

- This worked and fixed the issue.
89. Can you explain any difficult issue you have faced recently on High Availability Groups?

Ans:

Sure! We are configuring ALWAYSON AG on SQL Server 2014.

We have taken backup from Primary replica and restored on secondary replica

When we are trying to add secondary replica to availability group to our surprise SQL Server got shut down and we found the error message:

(Error: 3449, Severity: 21, State: 1.

SQL Server must shut down in order to recover a database (database ID 1). The database is either a user database that could not be shut down or a system database. Restart SQL

Cause:

We did RCA and found the below.

- Service broker is enabled at Primary Replica
- We have taken a full backup from Primary Replica
- Restored on Secondary Replica where Service Broker is not enabled
- When we try to add secondary replica to AG, Service Broker is enabled, the same GUID on availability database is detected which causes an silent error 9772:
 - “The Service Broker in database “<dbname>” cannot be enabled because there is already an enabled Service Broker with the same ID.”
 - This results into error 3449 and shut down the SQL Server unexpectedly.

Solution:

This has been fixed by applying the CU1 on SQL Server 2014.

90. Replica is in “resolving” status? What does it mean?

Ans:

A replica is into “RESOLVING” state when an auto failover is not successful.

Additionally the availability databases reports that they are in non-synchronizing state and not accessible.

91. What are the top reasons that cause an unsuccessful failover?

Ans:

- Auto failovers in a specific period may crossed the value “Maximum Failures in the Specified Period”
- Insufficient NT Authority\SYSTEM account permissions

- The availability databases are not in a SYNCHRONIZED state
92. Create Availability Group Fails with Error 35250 'Failed to join the database'. Have you ever encountered this kind of error? What are the possible reasons and solutions for this error?

Ans:

Yes! I faced this failure when attempting to create an availability group. Here are the possible reasons that cause the failure:

Inbound Port 5022 Traffic is blocked: By default, ALWAYSON configures the database mirroring endpoints to use port 5022. Make sure the inbound traffic is enabled for this port on windows firewall and TELNET the server to make sure the server is listening on 5022.

Endpoint is not created or not started: Check the End Point status on all replicas using the query “select name, state_desc, port from sys.tcp_endpoints where type_desc='DATABASE_MIRRORING' ”. If End point is not available or not started, create an end point and start it.

- To create an endpoint:

```
Create endpoint [Hadr_endpoint] state=started  
as tcp(listener_port = 5022, listener_ip = all)  
for database_mirroring (role = all, authentication =  
windows negotiate, encryption = required algorithm aes)
```

- To Start an End Point:

```
Alter endpoint [Hadr_endpoint] state = started
```

Note: Sometimes sys.tcp_endpoints.state_desc may incorrectly report the endpoint as STARTED when it is not started. Try to

execute ALTER ENDPOINT and start it

Endpoint permissions issue:

If database mirroring endpoints are configured to use Windows authentication, ensure that the SQL Server instances hosting your availability replicas run with a SQL Server STARTUP account are domain accounts. Also make sure SQL Service account has CONNECT permission on the end_point.

```
GRANT CONNECT ON ENDPOINT::[Hadr_endpoint] TO  
[DOMAINACCOUNT$]
```

SQL Server is not listening on port 5022:

To determine if SQL Server is listening on port 5022, review the SQL Server error log. You should find the following message(s) in the SQL Server error log:

Server is listening on ['any' <ipv6> 5022].

Server is listening on ['any' <ipv4> 5022].

SQL Server may not be able to listen on port 5022 if another application is already listening on the port. Run ‘netstat -a’ to determine what application is using the port:

93. Any idea about error 19471 error while creating Microsoft SQL Server 2012 ALWAYSON availability group listener?

Ans:

There might be two common reasons that cause to occur 19471. In this case SQL Server doesn't give the much information instead we can get it from cluster log.

We cannot create a group listener because of domain policy restriction: For example an Active Directory policy is defined that allows authenticated users in a domain that are assigned the “Add workstations to a domain” user permission and can create up to 20 computer accounts in the domain. We received

the error 19471 because we have exceeded the limit.

You cannot create a listener because of cluster name account permissions: This might be the reason when the cluster name account is not having sufficient permissions to create computer object on Active Directory.