

Always on is new high availability cluster feature in SQL server 2012. Always on uses clustering technology for Failover and also Mirroring in keeping multiple Mirrored copies of the databases. As of now, two features fall under the umbrella of AlwaysOn. These two features support high availability and disaster recovery for SQL Server databases:

- **SQL Server AlwaysOn Failover Cluster Instances (FCIs)**
- **SQL Server AlwaysOn Availability Groups (AGs)**

In Always on we can failover multiple Databases at the same time using Availability groups.

- AlwaysOn provides ability to have multiple Replicas which is not available in DB Mirroring. We can have one Primary and Secondary replicas with AlwaysOn. The Secondary replicas configured for read-access we can use for reporting purpose, backups etc.
- AlwaysOn also have 2 Availability modes.
  1. **Synchronous** – Provides zero data loss
  2. **Asynchronous** – Provides better performance, chance of data loss.

For each Replica we can configure separate mode.

## Components of AlwaysOn

1. **Availability Group:** - Availability group is a container to group databases. AlwaysOn availability groups contain multiple databases all of which can be automatically failover as single unit.
2. **Availability Replica:** - Server that contains the Availability group which can use for failover.
3. **Primary Replica:** - Server that holds read-write copy of the databases contained within Availability group.
4. **Secondary Replica:** - Secondary server that holds copy of primary replica database, only read option is available here is on Availability groups.

## How AlwaysOn works

AlwaysOn features based on Windows cluster services only. Availability groups are grouping databases with a virtual name and IP address that act as a single unit for users to access. If a server fails within an availability group the entire group is failover to the secondary replica of secondary server.

To create databases we need SQL server to be installed on a server under Windows clustered. SQL server Clustering does not need in AlwaysOn. Primary replica sends Transaction log

record of each primary database to very secondary database. The secondary replica applies changes on secondary database. Data synchronization occurs between primary and secondary databases. Windows Failover clusters monitor this resource group to evaluate health of primary replica.

The Quorum for AlwaysOn availability group resides on all nodes of cluster which helps in failover. There is no witness role in AlwaysOn availability groups.

### **Types of failovers are similar to Mirroring**

- **Synchronous commits supports planned and manual failover. Automatic failover and no data loss.**
- **A Synchronous commits supports only forced failover chance of data loss.**

### **Always on Availability group is superior to SQL server clustering**

- Configuration, Deployment, Installation is relatively simple comparing to normal Clustering
- All Replicas (Nodes) will have a copy of databases there is no shared storage hence we can avoid single point of failure.
- We have readable secondary replica hence we can distribute read-only load to secondary node and Read-write load to primary hence better utilization of both servers.

### **Always on is superior to Database Mirroring**

- We have up to 4 mirrored instances replicas here
- We can use contribution of synchronous commit mode for some databases and A synchronous commit mode for other databases at the same time not possible in mirroring.
- Synchronous mode for high availability (Automatic Failover) and A Synchronous mode for disaster recovery purpose.
- Backup options can be performed on secondary replica databases.
- Databases are always in recovery mode in mirroring here secondary replica database are readable, hence we can take load from primary replica.

### **Prerequisites-for-installing-sql-server-AlwaysOn**

- Failover Cluster enabled on both Windows Machine
- AlwaysOn Availability Group is an Enterprise Edition feature
- Same SQL Server collation for all replicas
- Take a Full Backup of primary server node
- Transaction log backup
- Restore Full and Transaction log backup on secondary node with Norecovery

**Step 1: Take a Full Backup of the Database in the primary replica.**

Backup database Mohsin to disk = 'D:\Backup\Mohsin.bak'

**Step 2:** Backup the Transactional Log of the database.

Backup Log Mohsin to disk = 'D:\Backup\Mohsin.trn'

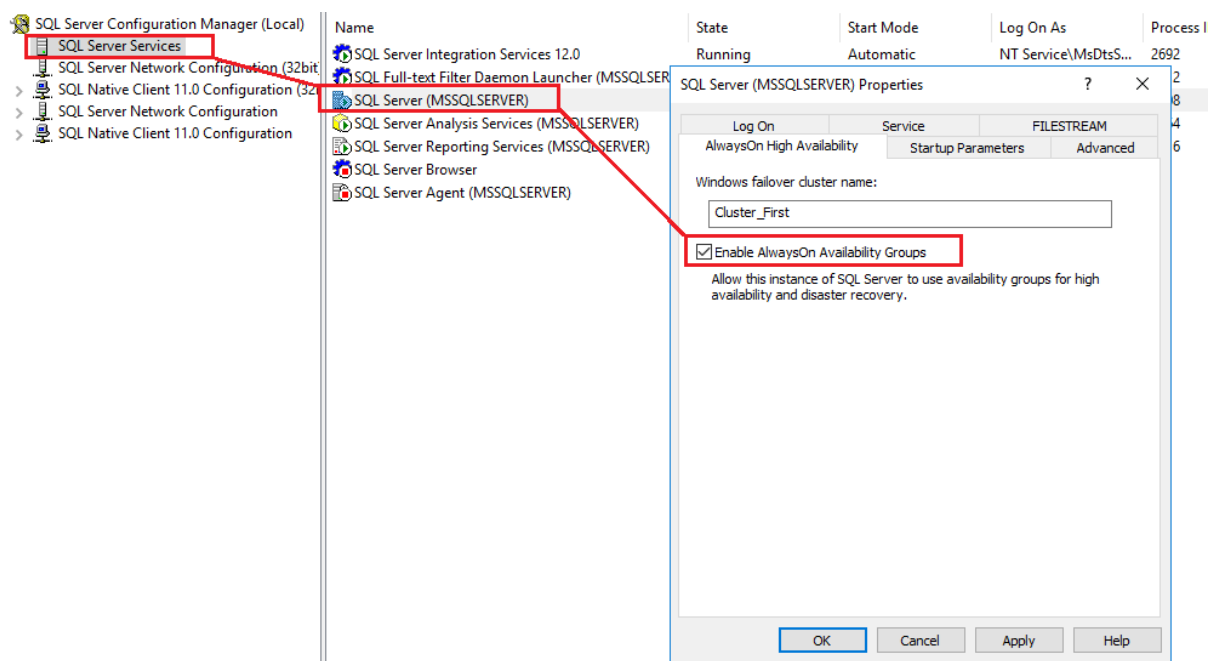
**Step 3:** Restore the full backup to the secondary Replica server with the **NORECOVERY** option.

Restore database Mohsin from disk = 'D:\Backup\Mohsin.bak' with NORECOVERY

**Step 4:** Restore log backup also with the **NORECOVERY** option.

Restore Log Mohsin from disk = 'D:\Backup\Mohsin.trn' with NORECOVERY

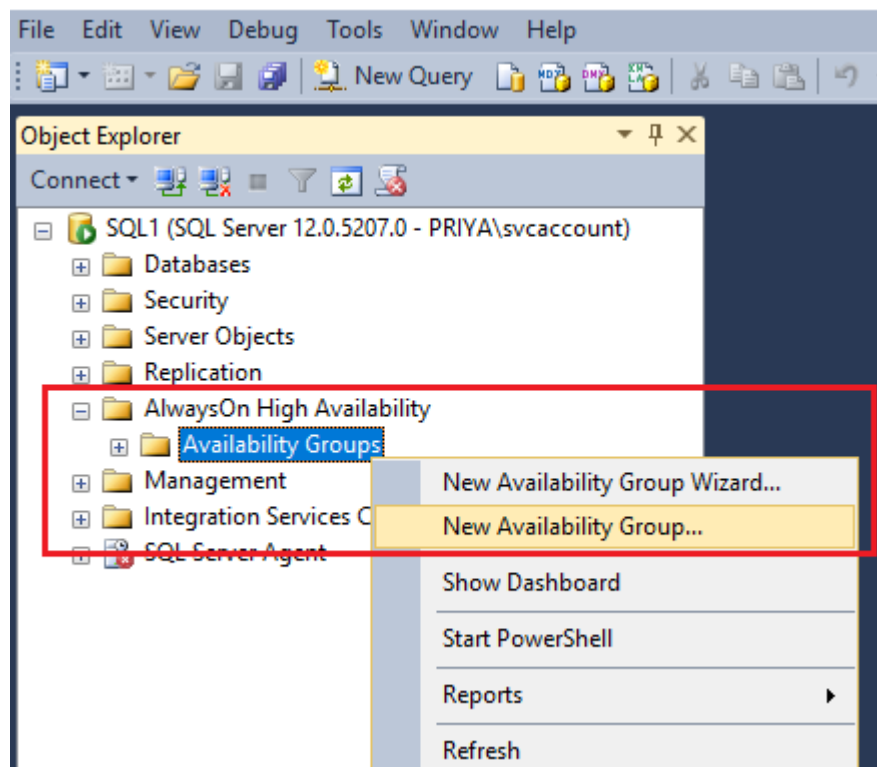
**Step 5:** To enable AlwaysOn feature, **Open SQL Server Configuration Manager, Right-click SQL Server instance and go to properties, Select AlwaysOn High Availability tab section and tick checkbox for Enable AlwaysOn Availability Groups.**



**STEP 6:** Once we enabled, Apply and Click **OK**. Restart the SQL Server services.

### Configuring Availability Group

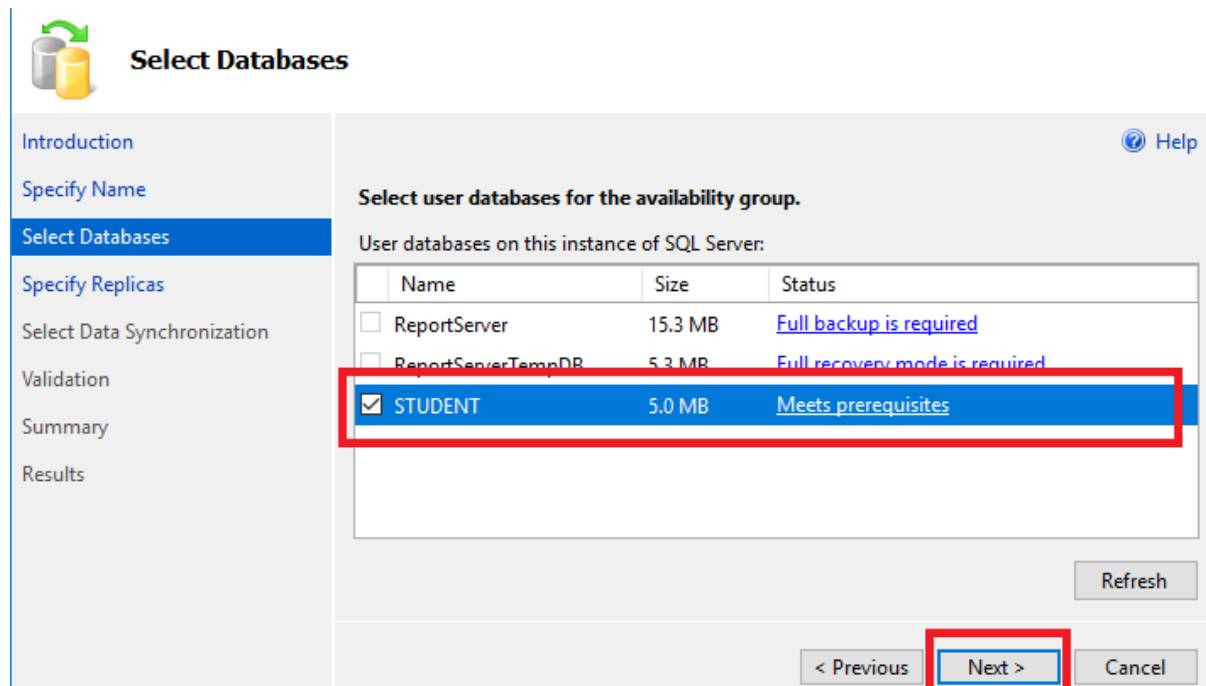
**STEP 7:** we will able to use SQL Server **AlwaysOn High Availability** feature. Right-click **Availability Groups** and Click **New Availability Group**.



**STEP 8:** Specify **Availability Group name** and click **Next**.

A screenshot of the 'Specify Availability Group Name' wizard in SQL Server Enterprise Manager. The wizard has a left-hand navigation pane with steps: 'Introduction', 'Specify Name' (current step), 'Select Databases', 'Specify Replicas', 'Select Data Synchronization', 'Validation', 'Summary', and 'Results'. The main area of the wizard is titled 'Specify an availability group name.' and contains a text box labeled 'Availability group name:' with the value 'AG1' entered. At the bottom right, there are three buttons: '< Previous', 'Next >' (which is highlighted with a red box), and 'Cancel'. A 'Help' icon is located in the top right corner.

**STEP 9:** In the **Select Databases** page select what database you want to include in **availability group**. A **full backup** is required to meet the requirements. Press **Next** to continue.



**Select Databases**

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

Help

Select user databases for the availability group.

User databases on this instance of SQL Server:

	Name	Size	Status
<input type="checkbox"/>	ReportServer	15.3 MB	<a href="#">Full backup is required</a>
<input type="checkbox"/>	ReportServerTempDB	5.3 MB	<a href="#">Full recovery mode is required</a>
<input checked="" type="checkbox"/>	STUDENT	5.0 MB	<a href="#">Meets prerequisites</a>

Refresh

< Previous **Next >** Cancel

Next, **Specify Replica** step has four sections. **Replica, Endpoint, Backup Preferences, and Listener.**

**Replica**– Replica is a server. There are one primary replica and multiple replicas. In SQL server 2012, **it supports up to 4 secondary replicas, while in SQL Server 2014, it supports up to 8 replicas.**

The primary replica is primary source server or production server. A secondary replica is a server which maintains a backup copy of the primary server availability database. On the Primary replica, it allows to perform Read and write operations while on the secondary replica only read operations.

In this case, SQL1 is our primary replica and SQL2 is a secondary replica. So, we will add SQL2 as a secondary replica as shown in the following fig.

**Initial Role** – It specifies the role of replica whether Primary or secondary.

**Automatic Failover** – It failovers functional role from primary replica to secondary replica by an automated method with no data loss.

**Synchronous Commit** – In the Synchronous-commit mode, failover can happen by either Automatic or manual way with no data loss. So, in this case, we will use synchronous commit with automatic failover.

**STEP 9:** Once clicked **Add Replica**, connect to the server, Select SQL2 and Connect.

New Availability Group

### Specify Replicas

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

Specify an instance of SQL Server to host a secondary replica.

Replicas Endpoints Backup Preferences Listener

Availability Replicas:

Server Instance	Initial Role	Automatic Failover (Up to 2)	Synchronous Commit (Up to 3)	Readable Secondary
SQL1	Primary	<input type="checkbox"/>	<input checked="" type="checkbox"/>	No

< >

Add Replica... Add Azure Replica... Remove Replica

**Summary for the replica hosted by SQL1**

**Replica mode:** Asynchronous commit  
This replica will use asynchronous-commit availability mode and support only forced failover (with possible data loss).

**Readable secondary:** No  
In the secondary role, this availability replica will not allow any connections.

< Previous Next > Cancel

Connect to Server

### Microsoft SQL Server 2014

Server type: Database Engine

Server name: SQL2

Authentication: Windows Authentication

User name: PRIYA\svcaccount

Password:

☐ Remember password

Connect Cancel Help Options >>

Once we add secondary server SQL2 as a secondary replica, the primary role of SQL2 is secondary.



## Specify Replicas

Introduction

Specify Name

Select Databases

Specify Replicas

Select Data Synchronization

Validation

Summary

Results

Help

Specify an instance of SQL Server to host a secondary replica.

Replicas Endpoints Backup Preferences Listener

Availability Replicas:

Server Instance	Initial Role	Automatic Failover (Up to 2)	Synchronous Commit (Up to 3)	Readable Seco
SQL1	Primary	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	No
SQL2	Secondary	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Yes

&lt; &gt;

Add Replica...

Add Azure Replica...

Remove Replica

## Summary for the replica hosted by SQL2

**Replica mode:** Synchronous commit with automatic failover

This replica will use synchronous-commit availability mode and support both automatic failover and manual failover.

**Readable secondary:** Yes

In the secondary role, this availability replica will allow all connections for read access, including

&lt; Previous

Next &gt;

Cancel

**STEP 10:** In the Endpoint section, it will show configured endpoint URL, verify that the port number value is 5022, endpoint name, and the service account name for SQL1 and SQL2



## Specify Replicas

Introduction

Specify Name

Select Databases

Specify Replicas

Select Data Synchronization

Validation

Summary

Results

Specify an instance of SQL Server to host a secondary replica.

Replicas Endpoints Backup Preferences Listener

Endpoint values:

Server Name	Endpoint URL	Port Number	Endpoint Name	Encrypt Data	SQL Server Service Account
SQL1	TCP://sql1.priya.com:5022	5022	Hadr_endpoint	<input checked="" type="checkbox"/>	PRIYA\svcaccount
SQL2	TCP://sql2.priya.com:5022	5022	Hadr_endpoint	<input checked="" type="checkbox"/>	PRIYA\svcaccount

Status

**Backup Preferences** – Backup preferences indicate the backup location of the **availability group**. It provides four backup options.

- **Prefer Secondary** - Will backup on secondary with highest priority. If no secondary's are available, it will backup on primary.
- **Secondary Only** - Will backup on secondary with highest priority. If no secondary's are available, no backups will occur.
- **Primary** - Backups will occur on the primary only, whichever instance that happens to be at the time of the backup.
- **Any Replica** - Looks just at the backup priority and backups on the replica with the highest priority

On the secondary replica, we can only take a log backup, and Copy-only backup, Differential, and Full back up are not allowed on a secondary replica. While on the primary replica, we can perform Full, log, and Differential backups.

**STEP 11:** In the **Backup Preferences** Tab ensure **Prefer Secondary** radio button is selected.

The screenshot shows the 'New Availability Group' wizard in SQL Server Enterprise Manager. The 'Specify Replicas' step is active, and the 'Backup Preferences' tab is selected. The 'Prefer Secondary' radio button is selected, and the 'Backup Preferences' tab is highlighted with a red box. The 'Prefer Secondary' option is also highlighted with a red box.

**Specify an instance of SQL Server to host a secondary replica.**

Replicas | Endpoints | **Backup Preferences** | Listener

Where should backups occur?

- ☒ **Prefer Secondary**  
Automated backups for this availability group should occur on a secondary replica. If there is no secondary replica available, backups will be performed on the primary replica.
- ☐ **Secondary only**  
All automated backups for this availability group must occur on a secondary replica.
- ☐ **Primary**  
All automated backups for this availability group must occur on the current primary replica.
- ☐ **Any Replica**  
Backups can occur on any replica in the availability group.



**STEP 12:** Listener is a server through which nodes get to communicate with each other. The Listener has all information about availability group. We need to select **Create Availability group** listener option and specify Name, Port and static IP. **Press Next to continue.**



## Specify Replicas

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

Specify an instance of SQL Server to host a secondary replica.

Replicas Endpoints Backup Preferences **Listener**

Specify your preference for an availability group listener that will provide a client connection

☐ Do not create an availability group listener now  
You can create the listener later using the Add Availability Group Listener dialog.

☒ **Create an availability group listener**  
Specify your listener preferences for this availability group.

Listener DNS Name: AGLName

Port: 16333

Network Mode: Static IP

Subnet	IP Address
10.0.0.0/8	10.0.0.4

< Previous **Next >** Cancel

**STEP 13:** select join only and click next

New Availability Group

Select Initial Data Synchronization

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

Select your data synchronization preference.

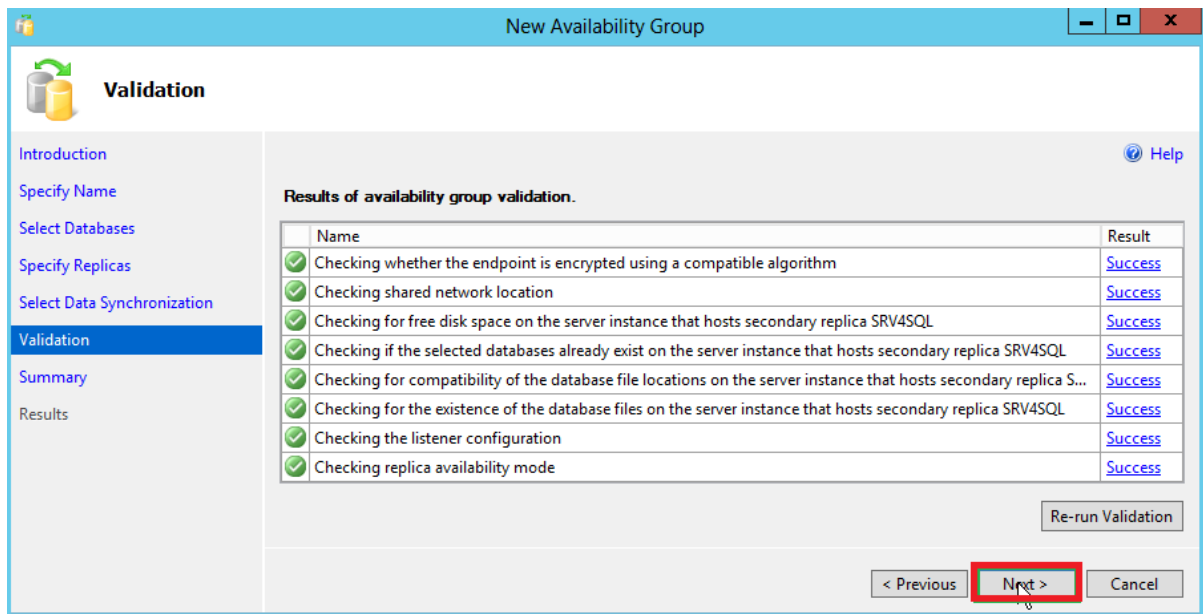
☐ Full  
Starts data synchronization by performing full database and log backups for each selected database. These databases are restored to each secondary and joined to the availability group.  
Specify a shared network location accessible by all replicas:  
\\SQLONE\SQLBACKUPS Browse...

☒ **Join only**  
Starts data synchronization where you have already restored database and log backups to each secondary server. The selected databases are joined to the availability group on each secondary. This action will be skipped for Azure replicas.

☐ Skip initial data synchronization  
Choose this option if you want to perform your own database and log backups of each primary database.

< Previous **Next >** Cancel

**STEP 14:** In the **Validation** page, verify that all validation checks return successful results. Click **Next**.



**Validation**

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

Help

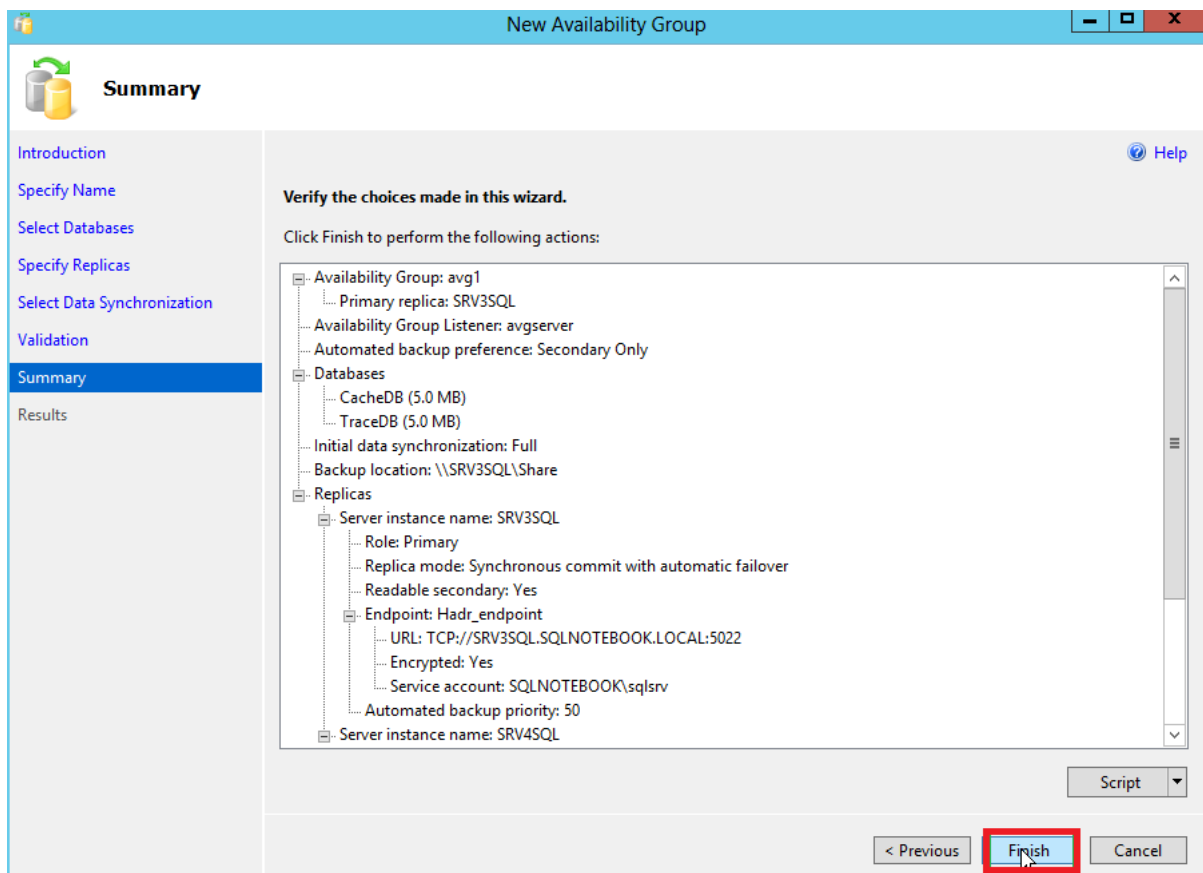
**Results of availability group validation.**

Name	Result
✓ Checking whether the endpoint is encrypted using a compatible algorithm	<a href="#">Success</a>
✓ Checking shared network location	<a href="#">Success</a>
✓ Checking for free disk space on the server instance that hosts secondary replica SRV4SQL	<a href="#">Success</a>
✓ Checking if the selected databases already exist on the server instance that hosts secondary replica SRV4SQL	<a href="#">Success</a>
✓ Checking for compatibility of the database file locations on the server instance that hosts secondary replica S...	<a href="#">Success</a>
✓ Checking for the existence of the database files on the server instance that hosts secondary replica SRV4SQL	<a href="#">Success</a>
✓ Checking the listener configuration	<a href="#">Success</a>
✓ Checking replica availability mode	<a href="#">Success</a>

Re-run Validation

< Previous **Next >** Cancel

**STEP 15:** Click **Finish**



**Summary**

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

Help

**Verify the choices made in this wizard.**

Click Finish to perform the following actions:

- Availability Group: avg1
  - Primary replica: SRV3SQL
- Availability Group Listener: avgserver
- Automated backup preference: Secondary Only
- Databases
  - CacheDB (5.0 MB)
  - TraceDB (5.0 MB)
- Initial data synchronization: Full
- Backup location: \\SRV3SQL\Share
- Replicas
  - Server instance name: SRV3SQL
    - Role: Primary
    - Replica mode: Synchronous commit with automatic failover
    - Readable secondary: Yes
  - Endpoint: Hadr\_endpoint
    - URL: TCP://SRV3SQL.SQLENTERBOOK.LOCAL:5022
    - Encrypted: Yes
    - Service account: SQLENTERBOOK\sqlsrv
  - Automated backup priority: 50
- Server instance name: SRV4SQL

Script

< Previous **Finish** Cancel

## Always On Scenarios

### ➤ How to add new database

- First create database on Primary server
- Take a full backup and log backup
- Restore both backup on secondary server with norecovery
- Right click on Availability Group then select add database Like :- Mumbai

### ➤ How to Remove database from Availability Group

- Right click on Database and click on remove

### ➤ How to Add data file on Availability Group Database on different location

- First remove database from Secondary server on Availability group
- Add data file on primary server and take one log backup
- Restore log backup with move command on secondary server
- On secondary server Availability group databases and right click on database and click join

### ➤ How to add data file on both Primary and Secondary server same folder

- First you create one folder on Primary server
- Same folder you can create on secondary server also
- Now Add data file on Primary server Then automatically reflect on secondary server

### ➤ How to install service pack

- First you suspend data movement on All the Availability group databases from secondary server
- After that you can install service pack on secondary server
- After installed service pack / click on resume data movement
- Command :- ALTER DATABASE Mohsin SET HADR RESUME

### ➤ How to check Sys Dash Board

- Right click on Availability group and click on dash board

### ➤ How to Failover GUI & Command

- GUI :- Go to Primary server & right click on Availability Group name and click failover
- Command:- Go to Secondary server and type command :- Alter availability group <Group name> failover

➤ **Set A Synchronous commit on Availability Group & Failover the group**

- In Primary server Right click on Availability group > Go to Properties > un Checked Synchronous commit > click on save OR ok.
- Right click on Availability Groups Failover & check the status

➤ **How to set Synchronous commit on secondary server**

- USE [master]  
ALTER AVAILABILITY GROUP MUMBAI  
MODIFY REPLICA ON 'Secondary Instance name' WITH (AVAILABILITY\_MODE = SYNCHRONOUS\_COMMIT)

➤ **change the disaster recovery replica node back to asynchronous commit**

- USE [master]  
ALTER AVAILABILITY GROUP MUMBAI  
MODIFY REPLICA ON 'Secondary Instance name' WITH (AVAILABILITY\_MODE = ASYNCHRONOUS\_COMMIT)

**DMV's**

- **To view the quorum model of the Windows cluster hosting the availability group, query the DMV**

sys.dm\_hadr\_cluster

- **To view the node votes, query the DMV**

sys.dm\_hadr\_cluster\_members

- **Confirm the synchronization status between the two locations**

sys.dm\_hadr\_availability\_replica\_states

- **How to use command for force failover**

ALTER AVAILABILITY GROUP MUMBAI FORCE\_FAILOVER\_ALLOW\_DATA\_LOSS