BASIC CONCEPTS & IMPLEMENTATION ON ALWAYS ON AVAILABILITY GROUPS

Prepared by: SAIFUL

Prepared for: Knowledge sharing with the DataSoft AML Team.

Date: 12.04.2021

Contents

ALWAYS ON AVAILABILITY GROUPS BASIC CONCEPTS

What is Always On Availability Groups

Always on is an enterprise-level high availability and disaster recovery solution that will replace the database mirroring feature.

Always On is a commercial name for a set of High Availability features that include Failover Cluster Instances and Availability Groups.

Always On Availability Group provides a high availability solution on the group level, where each group can contain any number of databases that can be replicated to multiple secondary servers known as Replicas.

It also supports to offload some read workloads to the secondary replica. We can also configure database backups from the secondary instance.

Fundamental Requirements

To support Always On availability groups feature, ensure that every computer that is to participate in one or more availability groups meets the following fundamental requirements:

1. Ensure that each computer is running Windows Server 2012 or later versions.

2. Each server instance must be running the same version of SQL Server and same SQL Server collation to participate in an Always On Availability Group.

3. All server instances that host an availability replica for the availability group must use the same SQL Server service account.

4. Ensure that each computer is a node in a WSFC.Ensure that the WSFC contains sufficient nodes to support your availability group configurations. A cluster node can host one replica for an availability group. The same node cannot host two replicas from the same availability group. The cluster node can participate in multiple availability groups, with one replica from each group.

5. Enable the Always On availability groups feature on each server instance that will host an availability replica for any availability group. On a given computer, you can enable as many server instances for Always On availability groups as your SQL Server installation supports.If you destroy and re-create a WSFC, you must disable and re-enable the Always On availability groups feature on each server instance that was enabled for Always On availability groups on the original cluster.

6. Ensure that the system is not a domain controller.Availability groups are not supported on domain controllers

7. Each server instance requires a database mirroring endpoint. Note that this endpoint is shared by all the availability replicas and database mirroring partners and witnesses on the server instance.If a server instance that you select to host an availability replica is running under a domain user account and does not yet have a database mirroring endpoint, the New Availability Group Wizard (or Add Replica to Availability Group Wizard) can create the endpoint and grant CONNECT permission to the server instance service account. However, if the SQL Server service is running as a built-in account, such as Local System, Local Service, or Network Service, or a non-domain account, you must use certificates for endpoint authentication, and the wizard will be unable to create a database mirroring endpoint on the server instance. In this case, we recommend that you create the database mirroring endpoints manually before you launch the wizard.

Always On Availability Groups Features

Always on feature have introduced in SQL Server 2012 version. Enhancement features of Always On Availability Groups in SQL Server (2012-2017) given bellow:

SQL Server 2012

1. Only 4 secondary replicas can be available in SQL server 2012 including one primary replica.

2. Only two automatic failovers will takes place in In SQL Server 2012.

3. Three synchronous mode replicas available in SQL server 2012.

SQL Server 2014

1. 8 Secondary Replicas are there in SQL Server 2014 including Primary.

2. Two automatic failovers will takes place in SQL server 2014.

3. Three synchronous replicas are there in SQL server 2014.

4. Read intent was introduced in SQL Server 2014.

Read intent routing command

```
ALTER AVAILABILITY GROUP [AG2]

MODIFY REPLICA ON N’WIN2012R2-VM1\GeoPITS_PR’

WITH
(PRIMARY_ROLE(READ_ONLY_ROUTING_LIST=(‘WIN2012R2-VM2\GeoPITS_SR1’,’WIN2012R2-VM3\GeoPITS_SR
2',’WIN2012R2-VM1\GeoPITS_PR’)));
```

In this read intent routing the read request will direct based on the order, which you have given in above command. Based on the above routing list all read request will direct to the ‘WIN2012R2-VM2\GeoPITS_SR1’ which is give in first in above routing list in case if these secondary replica is next available the read requests will direst to next replica which is second(WIN2012R2-VM3\GeoPITS_SR2) in routing list. In both first and second secondary replicas not available read requests will go next secondary replica based on routing list.

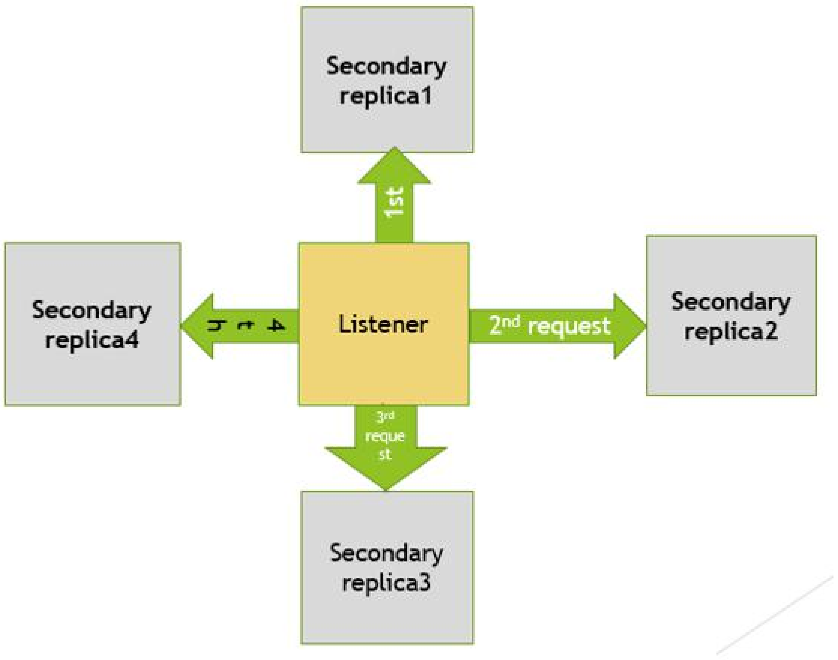
SQL server 2016

The Enhancements of ALWAYS ON in SQL server 2016 given below

1. Round-robin Load-balanced readable secondary’s.
2. Direct seeding of new DB replicas.
3. Supports Azure integration.
4. Always on Supports TDE.
5. Support for Distributed Transactions (DTC).
6. SQL Server Standard Edition support always on Availability group(Basic Availability Group).
7. ALWAYSON’ renamed as ‘ALWAYS ON’.

Round-robin load balancing in readable secondary’s:

In SQL server 2016 round-robin load balancing concept was introduced. Before SQL server 2016 there is no round-robin load balancing. Main use of the round-robin read requests will distributes equality to all secondary replicas as show in below figure.



Prior of SQL server 2016 the read request will connect to the secondary replica which will responds quickly for listener.

Automatic Seeding:

Automatic seeding is new enhancement of alwayson in SQL server 2016.

SQL Server performs a full database backup using Microsoft SQL Server Virtual Device Interface (VDI) full database.

This VDI database backup is streamed through the network to all available secondary replicas.

Secondary replica restores this streamed backup.

Once the database restoration is complete, it is added into the availability group.

GRANT CREATE ANY DATABASE permission needed.

Supports Azure:

In SQL Server 2016 Always on support AZURE for Secondary replica.

Recommended is secondary replica in Asynchronous mode.

Always on Supports TDE:

Now always on supports for Encrypted Databases.

However authentication requires when adding encrypted database to Availability Group.

Support for Distributed Transactions (DTC)

Always on supports the Distributed Transaction in SQL server 2016, now we register a resource manager per availability database. The resource manager works with DTC services to track distributed Transactions. Because of this now we can guarantee for integrity of a distributed Transaction.

To use DTC with Always on we require Windows Server 2012 with KB3090973.

We need to create Availability Group with the WITH DTC_SUPPORT =PER_DB.

Database level health detection failover:

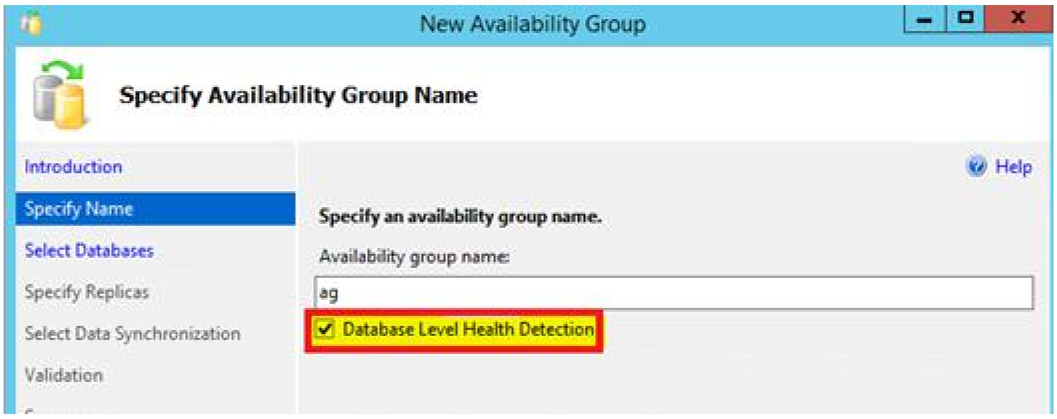
In SQL server 2016 always on one of the new enhancement is DATABSE level Health detection failover was introduced.in prior SQL server 2016 only Availability group failover takes place when the instance fails. From SQL server 2016 availability group failover takes place even on database fails or not available for long time.

We can enable this feature using below commands.

ALTER AVAILABILITY GROUP [AG1] SET (DB_FAILOVER = ON);

ALTER AVAILABILITY GROUP [AG2] SET (DB_FAILOVER = OFF);

We can see how to enable Database level Health detection for Availability group in below pictures for SQL server 2016 when creating new group.


New Availability Group

Specify Availability Group Name

Introduction
Specify Name
Select Databases
Specify Replicas
Select Data Synchronization
Validation

Help

Specify an availability group name.

Availability group name:

ag

[✓] Database Level Health Detection

SQL Server 2017

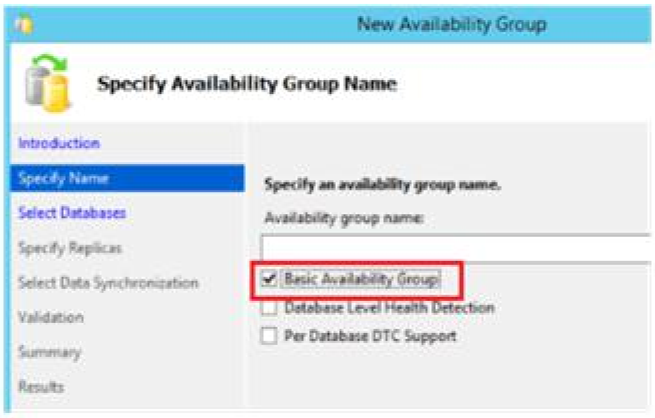
Read-scale Availability Group in SQL server 2017 and later we can create the availability group without windows failover cluster.

But it will not provides the high availability.

Basic Availability Group

Basic Availability group feature was introduced in SQL server 2016, which was supports SQL Server 2016 standard edition. Basic availability behaves like mirroring feature.

In below picture shows how to enable Basic availability in SQL Server 2016 and higher SQL server Standard edition.


New Availability Group

Specify Availability Group Name

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

Specify an availability group name.

Availability group name:

[✓] Basic Availability Group
[ ] Database Level Health Detection
[ ] Per Database DTC Support

Limitations of Basic Availability Group:

Secondary replica not allows read operations

No backup in secondary replica

Basic availability group supports only in standard editions in 2016 and later

Basic availability group supports failover environment for a single database.

In Basic availability group we have only on secondary replica.

Availability, Failure & Connection Mode

Overview of Failover

The following table summarizes which forms of failover are supported under different availability and failover modes. For each pairing, the effective availability mode and failover mode is determined by the intersection of the modes of the primary replica plus the modes of one or more secondary replicas.

Always On availability groups supports two availability modes-*asynchronous-commit mode* and *synchronous-commit mode*.

OVERVIEW OF FAILOVER

Failover form	**Asynchronous-commit mode**	**Synchronous-commit mode with manual-failover mode**	**Synchronous-commit mode with automatic-failover mode**
Automatic failover	No	No	Yes
Planned manual failover	No	Yes	Yes
Forced failover	Yes	Yes	Yes*****

*****If you issue a forced failover command on a synchronized secondary replica, the secondary replica behaves the same as for a manual failover.

Failover Sets

The forms of failover that are possible for a given availability group can be understood in terms of failover sets. A failover set consists of the primary replica and secondary replicas that support a given form of failover, as follows:

Automatic failover set (optional): Within a given availability group, a pair of availability replicas (including the current primary replica) that are configured for synchronous-commit mode with automatic failover, if any. An automatic failover set takes effect only if the secondary replica is currently SYNCHRONIZED with the primary replica.

Synchronous-commit failover set (optional): Within a given availability group, a set of two or three availability replicas (including the current primary replica) that are configured for synchronous-commit mode, if any. A synchronous-commit failover set takes effect only if the secondary replicas are configured for manual failover mode and at least one secondary replica is currently SYNCHRONIZED with the primary replica.

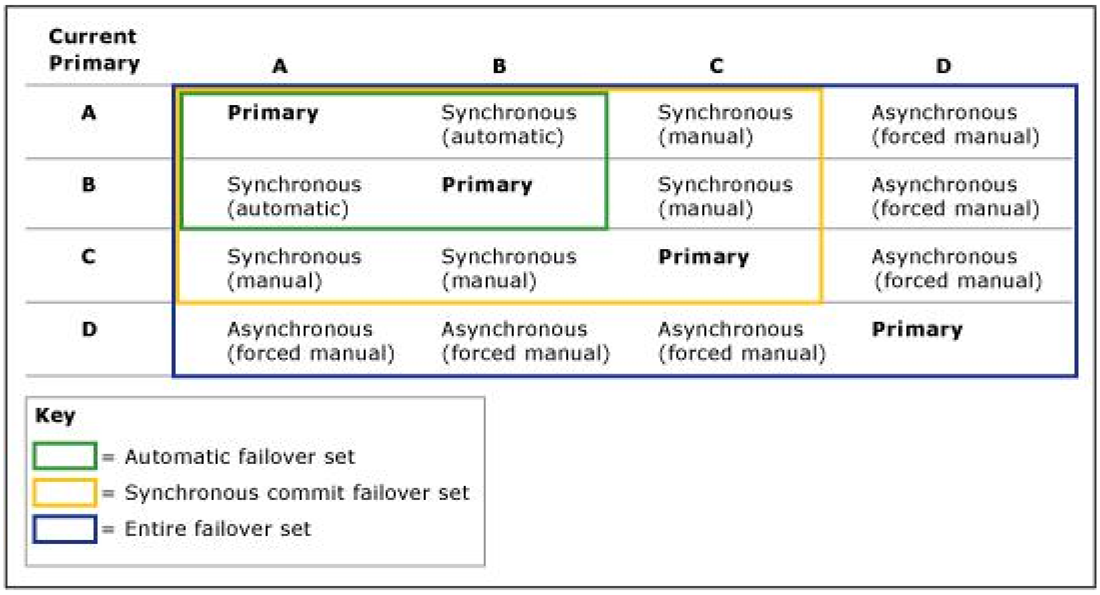
Entire failover set : Within a given availability group, the set of all availability replicas whose operational state is currently ONLINE, regardless of availability mode and of failover mode. The entire failover set becomes relevant when no secondary replica is currently SYNCHRONIZED with the primary replica.

When you configure an availability replica as synchronous commit with automatic failover, the availability replica becomes part of the automatic failover set. However whether the set takes effect depends the current primary. The forms of failover that are actually possible at a given time depends on what failover sets are currently in effect.

For example, consider an availability group that has four availability replicas, as follows:

	FAILOVER SETS
Replica	**Availability Mode and Failover Mode Settings**
A	Synchronous commit with automatic failover
B	Synchronous commit with automatic failover
C	Synchronous commit with planned manual failover only
D	Asynchronous commit (with only forced failover)

The failover behavior for each secondary replica depends on which availability replica is currently the primary replica. Basically, for a given secondary replica, the failover behavior is the worst case given the current primary replica. The following figure illustrates how the failover behavior of secondary replicas varies depending on the current primary replica, and whether it is configured for asynchronous-commit mode (with only forced failover) or synchronous-commit mode (with or without automatic failover).



Current Primary	**A**	**B**	**C**	**D**
A	**Primary**	Synchronous (automatic)	Synchronous (manual)	Asynchronous (forced manual)
B	Synchronous (automatic)	**Primary**	Synchronous (manual)	Asynchronous (forced manual)
C	Synchronous (manual)	Synchronous (manual)	**Primary**	Asynchronous (forced manual)
D	Asynchronous (forced manual)	Asynchronous (forced manual)	Asynchronous (forced manual)	**Primary**

Key

🟩 = Automatic failover set

🟨 = Synchronous commit failover set

🟦 = Entire failover set

ALWAYS ON AVAILABILITY GROUPS IMPLEMENTATION

Implementation Summary

1) Availability Groups Implementation with Failover Clustering (with/without readable secondary)

Step 1: Installing Windows Server Failover Clustering (WSFC) Feature

Step 2: Enabling Windows Server Failover Clustering (WSFC) Configuration for SQL Server

Step 3: Enabling SQL Server Always On Availability Groups Feature

Step 4: Creating and Configuring SQL Server Always On Availability Groups

Configure Read-only routing list in always on availability groups

2) Availability Groups Implementation without Failover Clustering (with readable secondary - Read Scale Availability Group new in SQL Server 2017)

It can be used for report server where no need to create failover cluster hence no High Availability and No Disaster Recovery only use secondary replica as read only report server to offload workload.

The setup is the same that you'd do for a regular availability group, with two distinctions:

1. Windows Server Failover Clustering is not needed
    1. The Availability Group manager does not have or need a cluster context
2. A new cluster_type option is present
    1. To use this, we want the cluster_type of "NONE"

Follow Step 3 and Step 4 from Regular Availability Group.

Implementation Details

1) Availability Groups Implementation with Failover Clustering (with/without readable secondary)

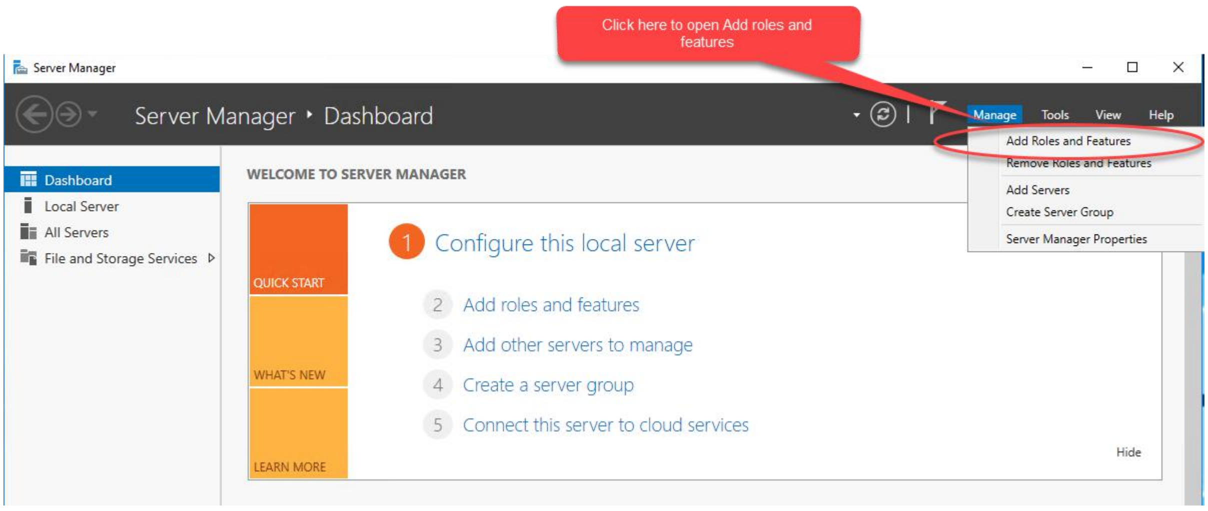
For demonstration purpose following components can be used:

Server/ Virtual Machine	**Host Name**	**Purpose**
Domain Controller	DC.Local	The domain controller is installed on this machine
Primary Replica	SQL01.DC.Local	This machine acts as a Primary replica in the Availability group
Secondary	SQL02.DC.Local	This machine acts as a Secondary replica in the Availability group. This replica is in a **Synchronous**

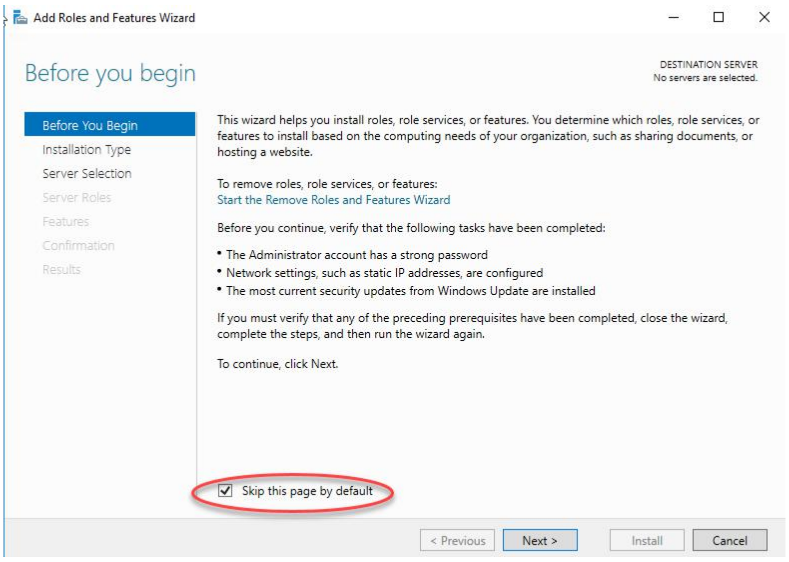
Replica		**commit** mode
Secondary Replica with	SQL03.DC.Local	This machine acts as a secondary replica in the Availability group. This replica is in an **Asynchronous commit** mode

Step 1: Installing Windows Server Failover Clustering (WSFC) Feature

To deploy AAG, first, we need to install a failover cluster feature on every node. To do that, connect to **SQL01.DC.Local** and open **Server Manager**. At the top side of the **Server Manager** dialog box, click on **Manage** and select **Add Roles and Features**. See the following image:


Click here to open Add roles and features
Server Manager
Server Manager ▸ Dashboard
Manage
Tools
View
Help
Add Roles and Features
Remove Roles and Features
Add Servers
Create Server Group
Server Manager Properties
Dashboard
Local Server
All Servers
File and Storage Services
WELCOME TO SERVER MANAGER
QUICK START
WHAT'S NEW
LEARN MORE
1 Configure this local server
2 Add roles and features
3 Add other servers to manage
4 Create a server group
5 Connect this server to cloud services
Hide

The **add/remove roles and features** wizard opens. On the first screen, you can see the information about this wizard. You can skip this screen by clicking on "**Skip this page by default**", then click on Next. See the following image:



This wizard helps you install roles, role services, or features. You determine which roles, role services, or features to install based on the computing needs of your organization, such as sharing documents, or hosting a website.

To remove roles, role services, or features:
Start the Remove Roles and Features Wizard

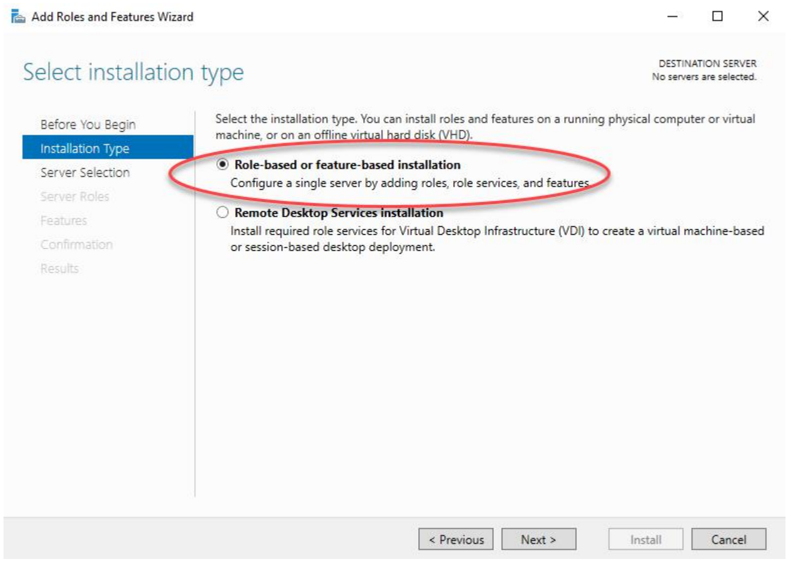
Before you continue, verify that the following tasks have been completed:

- The Administrator account has a strong password
- Network settings, such as static IP addresses, are configured
- The most current security updates from Windows Update are installed

If you must verify that any of the preceding prerequisites have been completed, close the wizard, complete the steps, and then run the wizard again.

To continue, click Next.

On the next screen, you can perform a role-based or feature-based installation. You can also choose remote desktop service based installation. Since we want to install a failover cluster feature on **SQL01.dc.local**, select role-based or feature-based installation. Select your option and click on **Next**. See the following image:


Add Roles and Features Wizard

Select installation type

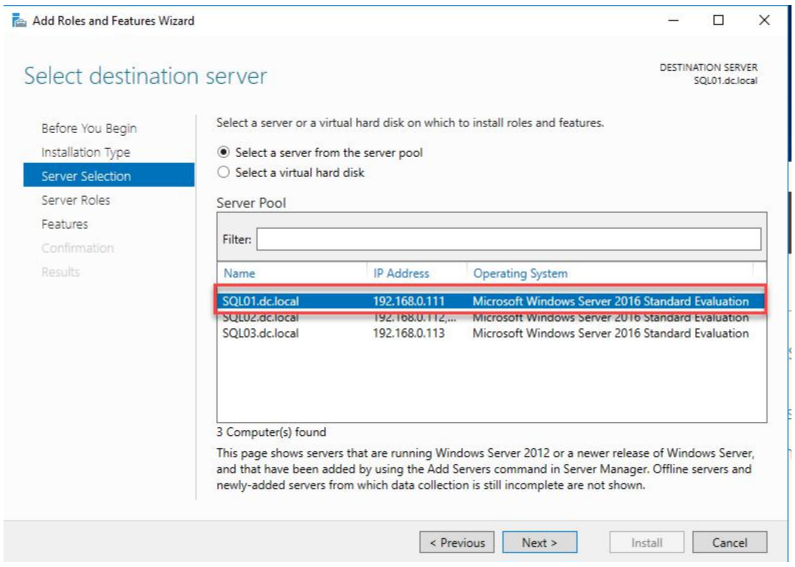
DESTINATION SERVER
No servers are selected.

Select the installation type. You can install roles and features on a running physical computer or virtual machine, or on an offline virtual hard disk (VHD).

● Role-based or feature-based installation
Configure a single server by adding roles, role services, and features.

○ Remote Desktop Services installation
Install required role services for Virtual Desktop Infrastructure (VDI) to create a virtual machine-based or session-based desktop deployment.

On the next screen, select a server on which you want to enable a failover cluster feature. Since we want to install it on **SQL01.Dc.Local**, choose **SQL01.Dc.Local** from the server pool and click on **Next**. See the following image:


Add Roles and Features Wizard

Select destination server

DESTINATION SERVER
SQL01.dc.local

Select a server or a virtual hard disk on which to install roles and features.

● Select a server from the server pool
○ Select a virtual hard disk

Server Pool

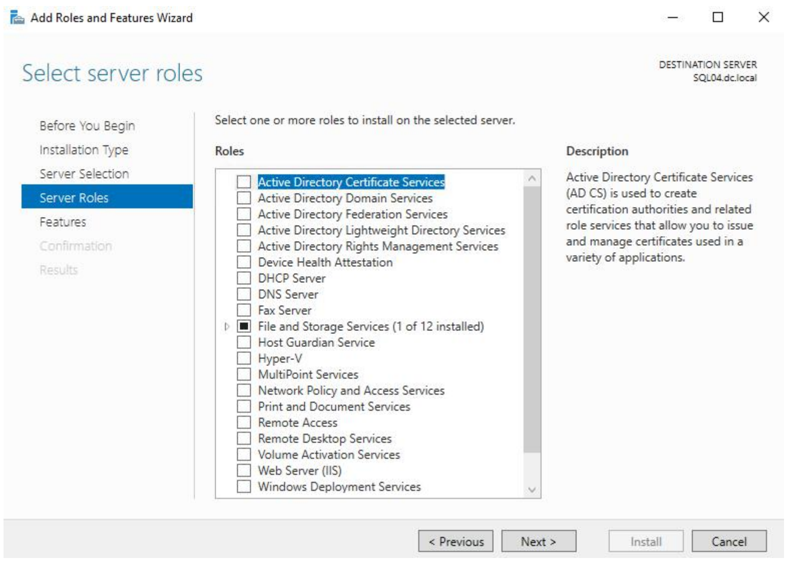
Filter:

Name	IP Address	Operating System
SQL01.dc.local	192.168.0.111	Microsoft Windows Server 2016 Standard Evaluation
SQL02.dc.local	192.168.0.112,...	Microsoft Windows Server 2016 Standard Evaluation
SQL03.dc.local	192.168.0.113	Microsoft Windows Server 2016 Standard Evaluation

3 Computer(s) found

This page shows servers that are running Windows Server 2012 or a newer release of Windows Server, and that have been added by using the Add Servers command in Server Manager. Offline servers and newly-added servers from which data collection is still incomplete are not shown.

On the next screen, you can review the list of server roles. The failover cluster is a feature, hence click on **Next**. See the following image:


Add Roles and Features Wizard

Select server roles

DESTINATION SERVER
SQL04.dc.local

Before You Begin
Installation Type
Server Selection
Server Roles
Features
Confirmation
Results

Select one or more roles to install on the selected server.

Roles

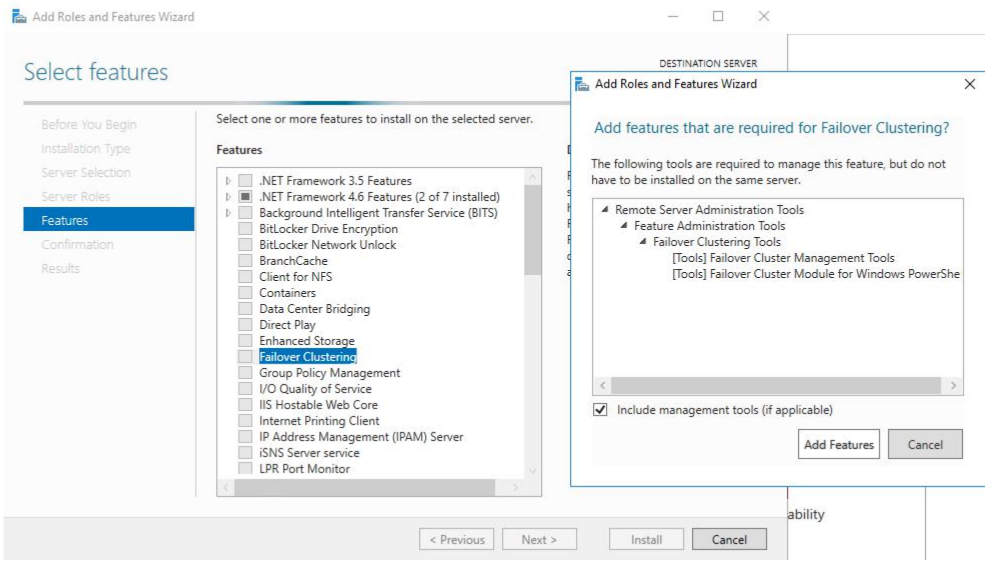
- [ ] Active Directory Certificate Services
- [ ] Active Directory Domain Services
- [ ] Active Directory Federation Services
- [ ] Active Directory Lightweight Directory Services
- [ ] Active Directory Rights Management Services
- [ ] Device Health Attestation
- [ ] DHCP Server
- [ ] DNS Server
- [ ] Fax Server
- [■] File and Storage Services (1 of 12 installed)
- [ ] Host Guardian Service
- [ ] Hyper-V
- [ ] MultiPoint Services
- [ ] Network Policy and Access Services
- [ ] Print and Document Services
- [ ] Remote Access
- [ ] Remote Desktop Services
- [ ] Volume Activation Services
- [ ] Web Server (IIS)
- [ ] Windows Deployment Services

Description

Active Directory Certificate Services (AD CS) is used to create certification authorities and related role services that allow you to issue and manage certificates used in a variety of applications.

< Previous | Next > | Install | Cancel

On the **Select Features** screen, select "**Failover Clustering**" from the list of features, click on **Add features,** and click on N**ext**. See the following image:


Add Roles and Features Wizard

Select features

DESTINATION SERVER

Before You Begin
Installation Type
Server Selection
Server Roles
Features
Confirmation
Results

Select one or more features to install on the selected server.

Features

- [ ] .NET Framework 3.5 Features
- [■] .NET Framework 4.6 Features (2 of 7 installed)
- [ ] Background Intelligent Transfer Service (BITS)
- [ ] BitLocker Drive Encryption
- [ ] BitLocker Network Unlock
- [ ] BranchCache
- [ ] Client for NFS
- [ ] Containers
- [ ] Data Center Bridging
- [ ] Direct Play
- [ ] Enhanced Storage
- [ ] Failover Clustering
- [ ] Group Policy Management
- [ ] I/O Quality of Service
- [ ] IIS Hostable Web Core
- [ ] Internet Printing Client
- [ ] IP Address Management (IPAM) Server
- [ ] iSNS Server service
- [ ] LPR Port Monitor

Add Roles and Features Wizard

Add features that are required for Failover Clustering?

The following tools are required to manage this feature, but do not have to be installed on the same server.

- Remote Server Administration Tools
  - Feature Administration Tools
    - Failover Clustering Tools
      - [Tools] Failover Cluster Management Tools
      - [Tools] Failover Cluster Module for Windows PowerShe

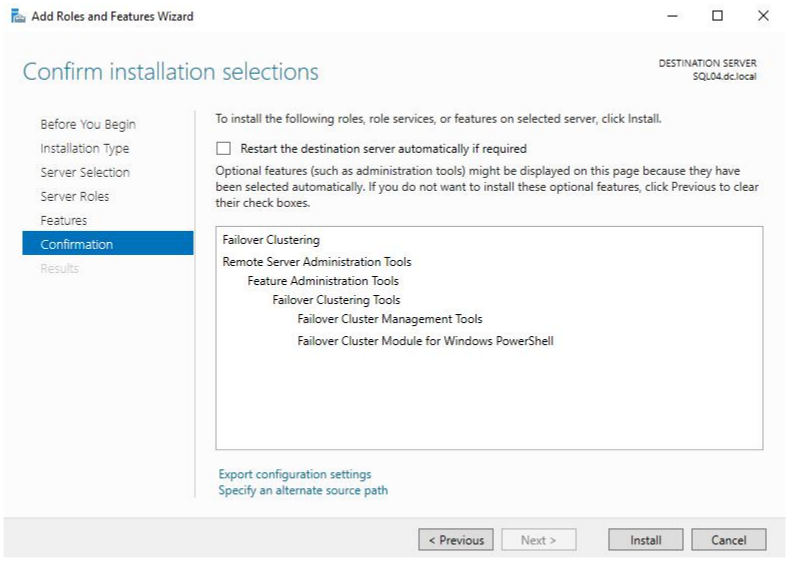
- [✓] Include management tools (if applicable)

Add Features | Cancel

ability

< Previous | Next > | Install | Cancel

On the next screen, you can review the summary of the feature installation. Click on **Install** and see the following image:


Add Roles and Features Wizard

Confirm installation selections

DESTINATION SERVER
SQL04.dc.local

Before You Begin
Installation Type
Server Selection
Server Roles
Features
Confirmation
Results

To install the following roles, role services, or features on selected server, click Install.

- [ ] Restart the destination server automatically if required

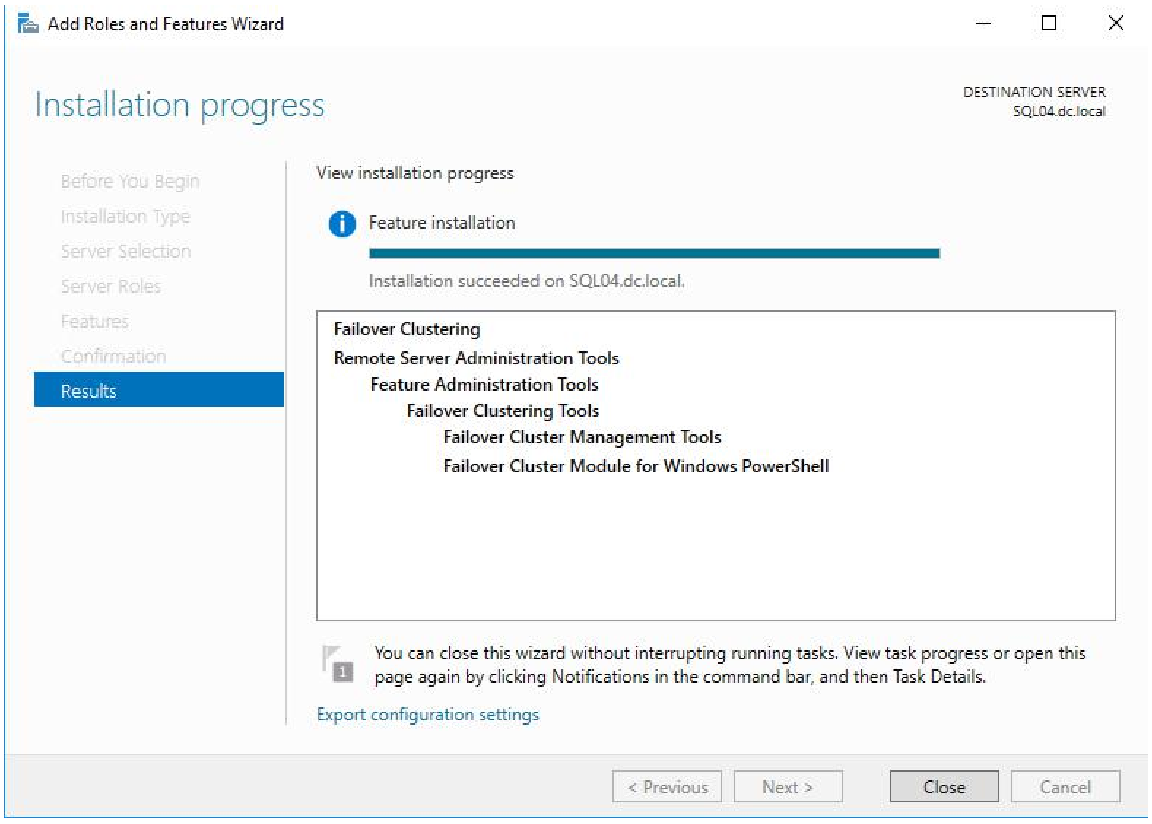
Optional features (such as administration tools) might be displayed on this page because they have been selected automatically. If you do not want to install these optional features, click Previous to clear their check boxes.

Failover Clustering
Remote Server Administration Tools
  Feature Administration Tools
    Failover Clustering Tools
      Failover Cluster Management Tools
      Failover Cluster Module for Windows PowerShell

Export configuration settings
Specify an alternate source path

< Previous | Next > | Install | Cancel

The failover clustering feature has been installed successfully. See the following image:



Installation progress

DESTINATION SERVER
SQL04.dc.local

View installation progress

Feature installation

Installation succeeded on SQL04.dc.local.

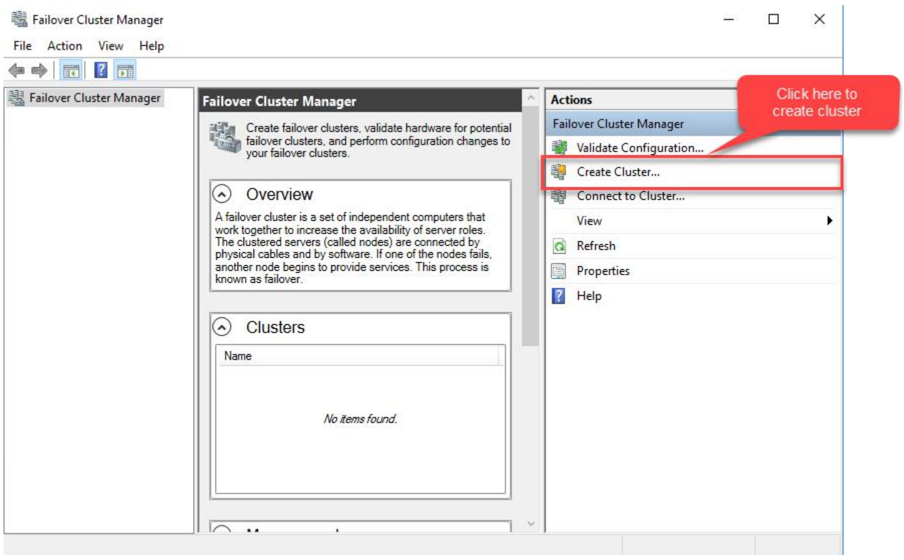
Failover Clustering
Remote Server Administration Tools
    Feature Administration Tools
        Failover Clustering Tools
            Failover Cluster Management Tools
            Failover Cluster Module for Windows PowerShell

You can close this wizard without interrupting running tasks. View task progress or open this page again by clicking Notifications in the command bar, and then Task Details.

Similarly, follow the above process to install the failover clustering feature on all the nodes.

Step 2: Enabling Windows Server Failover Clustering (WSFC) Configuration for SQL Server

To create a cluster, open the failover cluster manager and click on **Create Cluster**. See the following image:



Failover Cluster Manager

Create failover clusters, validate hardware for potential failover clusters, and perform configuration changes to your failover clusters.

Overview

A failover cluster is a set of independent computers that work together to increase the availability of server roles. The clustered servers (called nodes) are connected by physical cables and by software. If one of the nodes fails, another node begins to provide services. This process is known as failover.

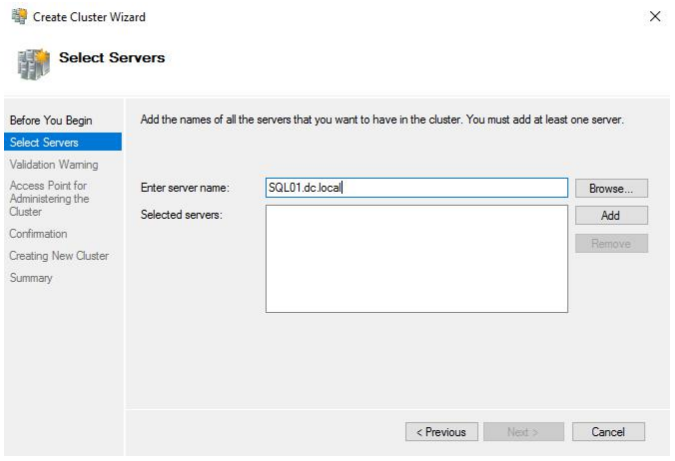
Clusters

Name

No items found.

Once the wizard creates a name, a new cluster opens. On the first screen, you can review the wizard details. Click on **Next**.

On the **Select Servers** screen, you need to add a list of nodes that you want to use to form a cluster. We will create a cluster using **SQL01.dc.local**, **SQL02.dc.local,** and **SQL03.dc.local**. To do that, first, enter **SQL01.dc.local** in the **enter server name** text box and click on **Add**. See the following image:

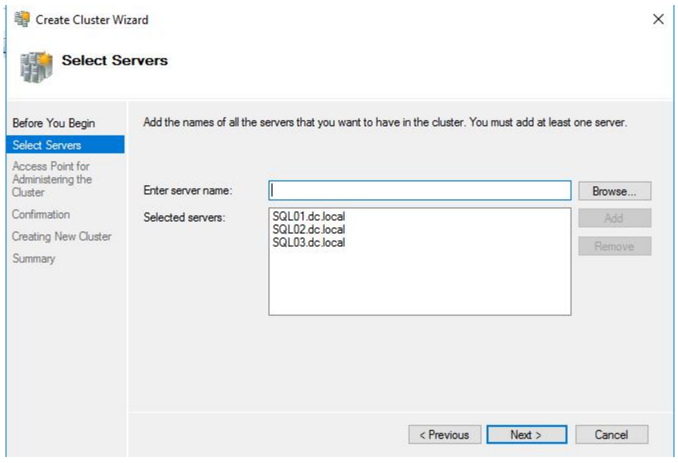


Add the names of all the servers that you want to have in the cluster. You must add at least one server.

Enter server name: SQL01.dc.local

Selected servers:

Similarly, add **SQL02.dc.local** and **SQL03.dc.local**, then click on next. See the following image:

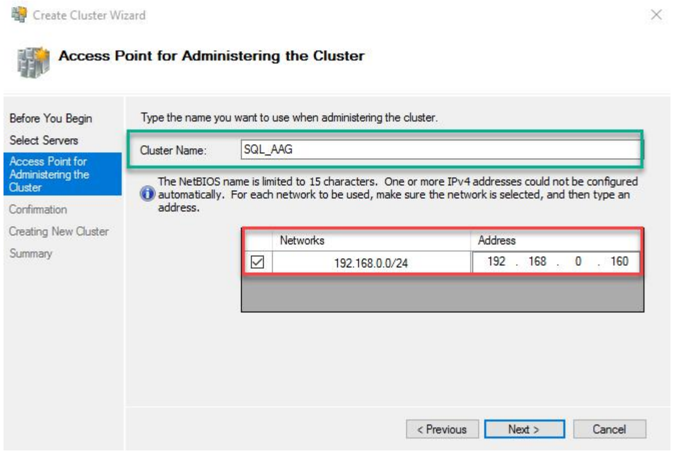


Add the names of all the servers that you want to have in the cluster. You must add at least one server.

Enter server name:

Selected servers: SQL01.dc.local
SQL02.dc.local
SQL03.dc.local

On the **Access Point to Administrating Cluster** screen, enter **Cluster Name** and IP address to access it. See the following image:


Create Cluster Wizard

Access Point for Administering the Cluster

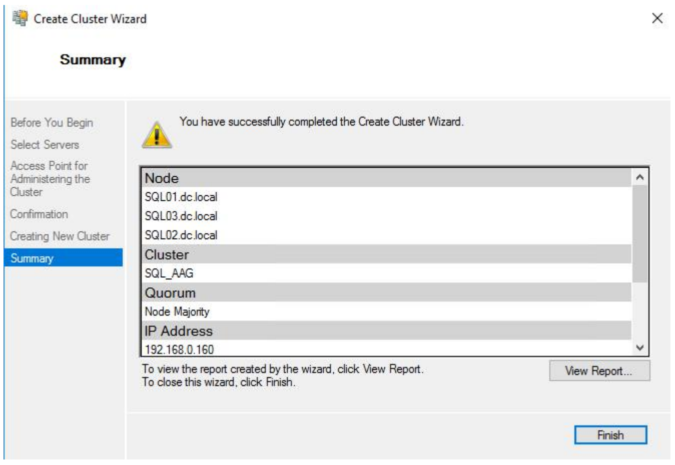
Type the name you want to use when administering the cluster.

Cluster Name: SQL_AAG

The NetBIOS name is limited to 15 characters. One or more IPv4 addresses could not be configured automatically. For each network to be used, make sure the network is selected, and then type an address.

	Networks	Address
☑	192.168.0.0/24	192 . 168 . 0 . 160

On the confirmation screen, review all the details and click on **Next**. The process of building a failover cluster will be started. Once the process completes, you can see the installation summary on the **Summary** screen. See the following image:


Create Cluster Wizard

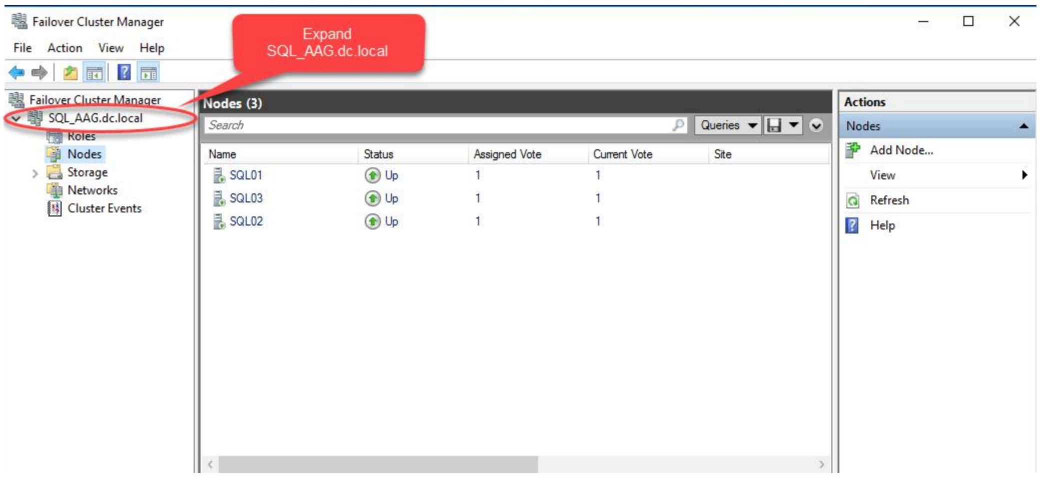
Summary

You have successfully completed the Create Cluster Wizard.

Node
SQL01.dc.local
SQL03.dc.local
SQL02.dc.local
Cluster
SQL_AAG
Quorum
Node Majority
IP Address
192.168.0.160

To view the report created by the wizard, click View Report.
To close this wizard, click Finish.

Once the cluster is created, you can review its configuration from the failover cluster manager. To view the details, connect to **SQL01.dc.local**, open **Failover Cluster Manager,** expand **SQL_AAG.dc.Local,** and select "**Nodes**" to view the underlying nodes. See the following image:



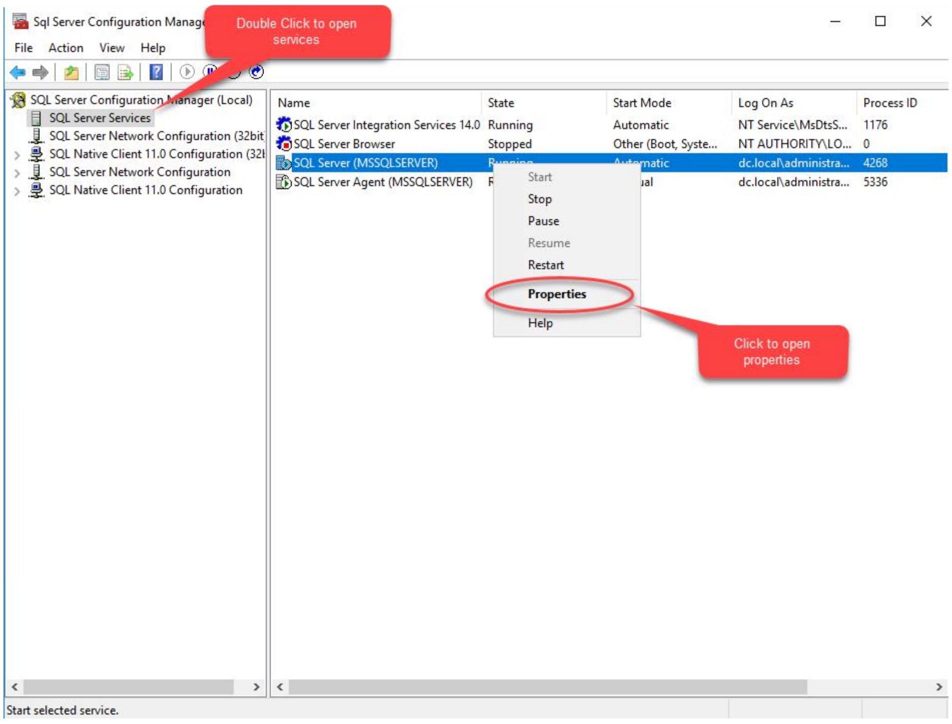
Nodes (3)

Name	Status	Assigned Vote	Current Vote	Site
SQL01	Up	1	1	
SQL03	Up	1	1	
SQL02	Up	1	1	

Step 3: Enabling SQL Server Always On Availability Groups Feature

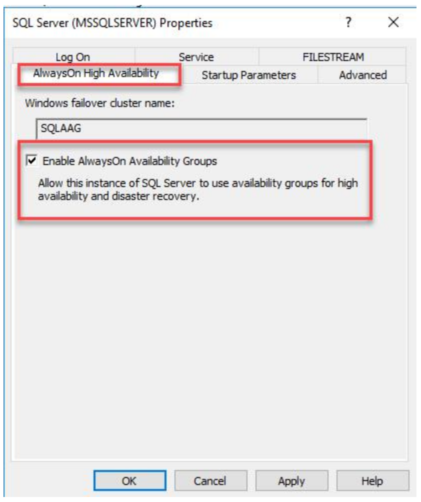
Once the cluster is created, we must install SQL Server 2017 on all the nodes.

After SQL Server is installed on the nodes, we need to enable Always On availability group features on all of them. To do that, connect to **SQ01.dc.local** –> Open **SQL Server 2017 Configuration Manager**, double-click on "***SQL Server Services***", and right-click on ***SQL Server (MSSQLSERVER)***. See the following image:


Sql Server Configuration Manager
File
Action
View
Help
Double Click to open services
SQL Server Configuration Manager (Local)
SQL Server Services
SQL Server Network Configuration (32bit
SQL Native Client 11.0 Configuration (32
SQL Server Network Configuration
SQL Native Client 11.0 Configuration
Start
Stop
Pause
Resume
Restart
Properties
Help
Click to open properties
Start selected service.

Name	State	Start Mode	Log On As	Process ID
SQL Server Integration Services 14.0	Running	Automatic	NT Service\MsDtsS...	1176
SQL Server Browser	Stopped	Other (Boot, Syste...	NT AUTHORITY\LO...	0
SQL Server (MSSQLSERVER)	Running	Automatic	dc.local\administra...	4268
SQL Server Agent (MSSQLSERVER)	R	ual	dc.local\administra...	5336

Now the SQL Server (MSSQLSERVER) dialog box opens. Once you see it, click on "**Always On High Availability**" and check the "**Enable Always On Availability group**" check-box. Click **Ok** to close the dialog box and restart SQL Service. See the following image:



Windows failover cluster name:

SQLAAG

☑ Enable AlwaysOn Availability Groups

Allow this instance of SQL Server to use availability groups for high availability and disaster recovery.

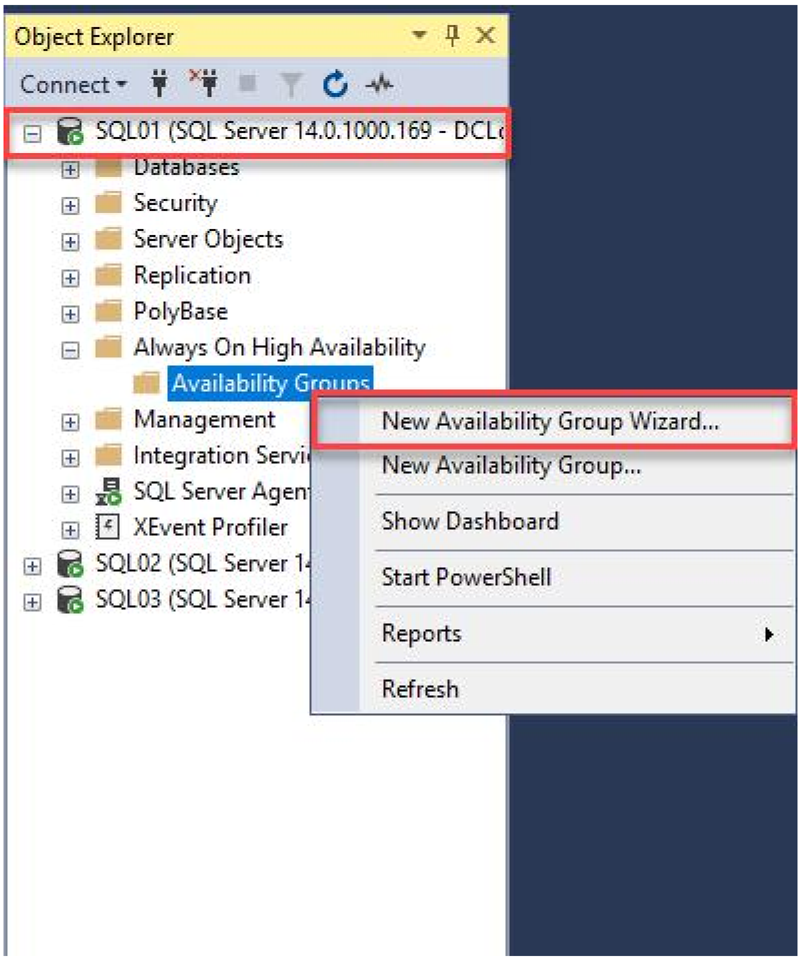
Similarly, we must enable these features on all the nodes. To do that, follow the above process for **SQL02.dc.local** and **SQL03.dc.local**.

Step 4: Creating and Configuring SQL Server Always On Availability Groups

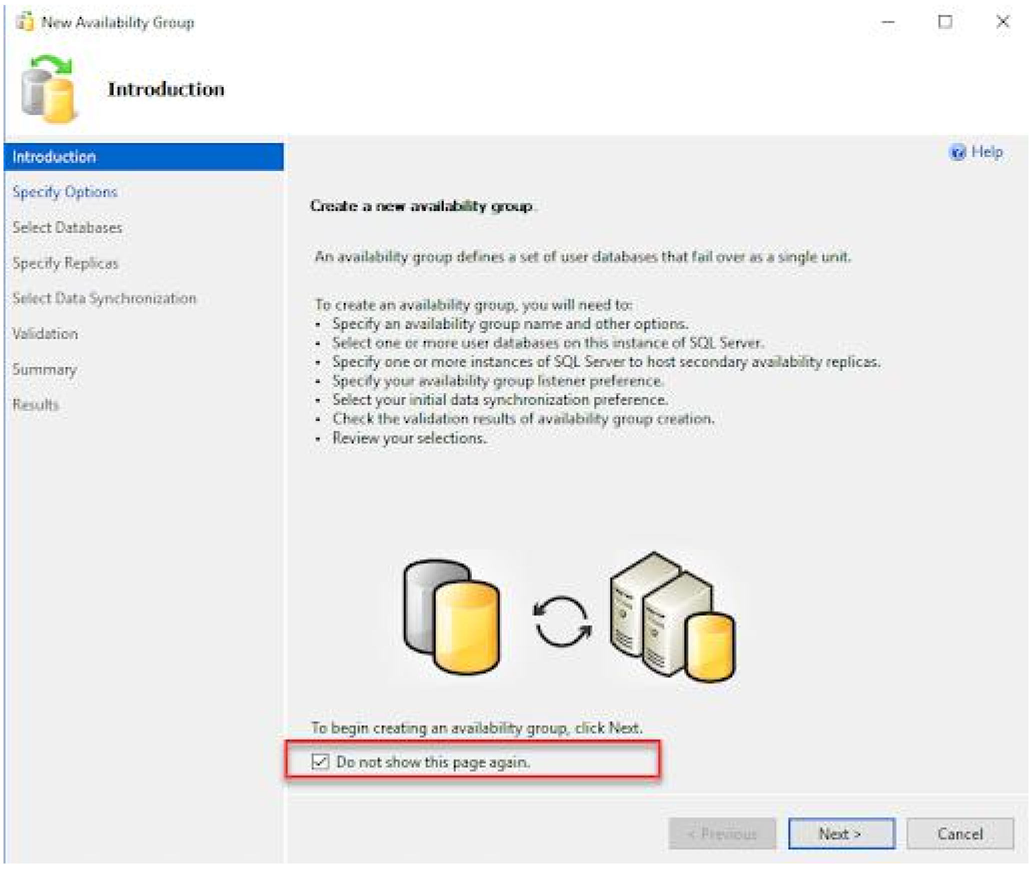
The deployment configurations look the following way:

Availability group name	Domain Name of the Server	Replica Type	Availability mode
AAG_Demo	SQL01.DC.Local	Primary	Synchronous commit mode
	SQL02.DC.Local	Secondary	Synchronous commit mode
	SQL03.DC.Local	Secondary	Asynchronous commit mode

First, connect to SQL01.DC.Local and open SQL Server management studio. In SSMS, connect to the database engine. In the object explorer window, expand **Always On High Availability**, Right-click on **Availability groups,** and select "**New Availability Group Wizard**." See the following image:



The first screen is Introduction, which provides the details of the availability group wizard and what tasks we can perform using it. If you do not wish to see this screen again, you can skip it by selecting "**Do not show this page again.**" Click **Next** to move on to the next screen. See the following screenshot:



Create a new availability group.

An availability group defines a set of user databases that fail over as a single unit.

To create an availability group, you will need to:
- Specify an availability group name and other options.
- Select one or more user databases on this instance of SQL Server.
- Specify one or more instances of SQL Server to host secondary availability replicas.
- Specify your availability group listener preference.
- Select your initial data synchronization preference.
- Check the validation results of availability group creation.
- Review your selections.

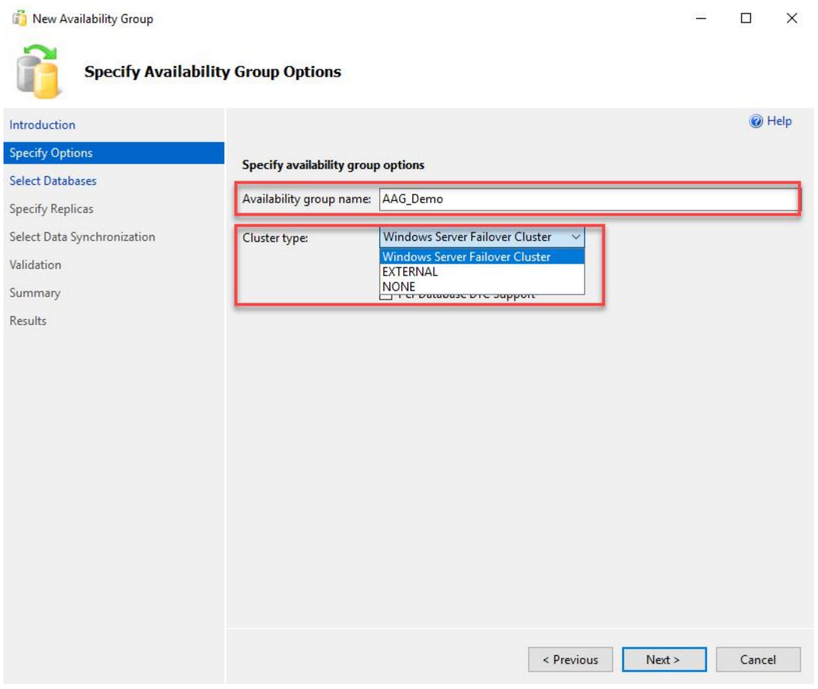
To begin creating an availability group, click Next.

☑ Do not show this page again.

On the specify availability group screen, enter the desired name of the availability group. In the "**Cluster Type**" drop-down menu, you can choose any of the following values:

1. **Windows Server Failover Cluster**: This option is used when you want to create an availability group using a traditional Windows Server failover cluster.
2. **External:** This option is used when you create an availability group on the Linux operating system. It uses the Linux operating system by integrating it with PACEMAKER (Linux Cluster resource manager).
3. **NONE:** This option is used when you do not wish to enable high availability option. It can be used for both Windows and Linux.

We are deploying AAG on a Windows cluster hence select "**Windows Server Failover Cluster**" from the cluster type drop-down box. Click **Next** to move on to the next screen. See the following screenshot:


New Availability Group

Specify Availability Group Options

Specify availability group options

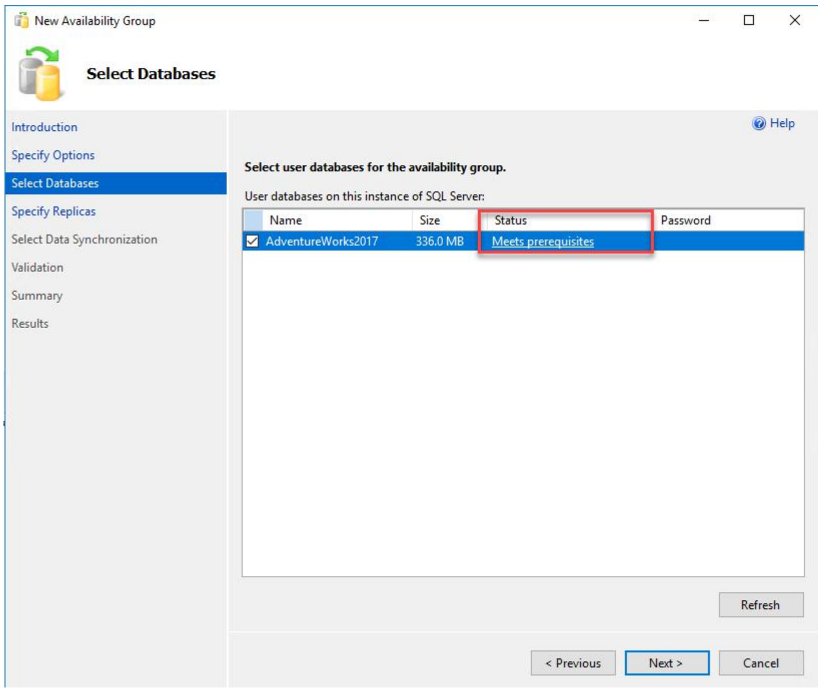
Availability group name: AAG_Demo

Cluster type: Windows Server Failover Cluster

On the **Select databases** screen, choose the databases you want to include in your availability group. The databases must fulfill the following prerequisites to be a part of the availability group:

1. The database must be in the Full Recovery Model.
2. The Full Backup of the database must be taken.

If the above pre-requisites are met, you can see "Meets Prerequisites" in the **Status** column of the grid. Choose the name of the database by clicking on the checkbox and click on **Next**. See the following image:


New Availability Group

Select Databases

Select user databases for the availability group.

User databases on this instance of SQL Server:

	Name	Size	Status	Password
✓	AdventureWorks2017	336.0 MB	Meets prerequisites	

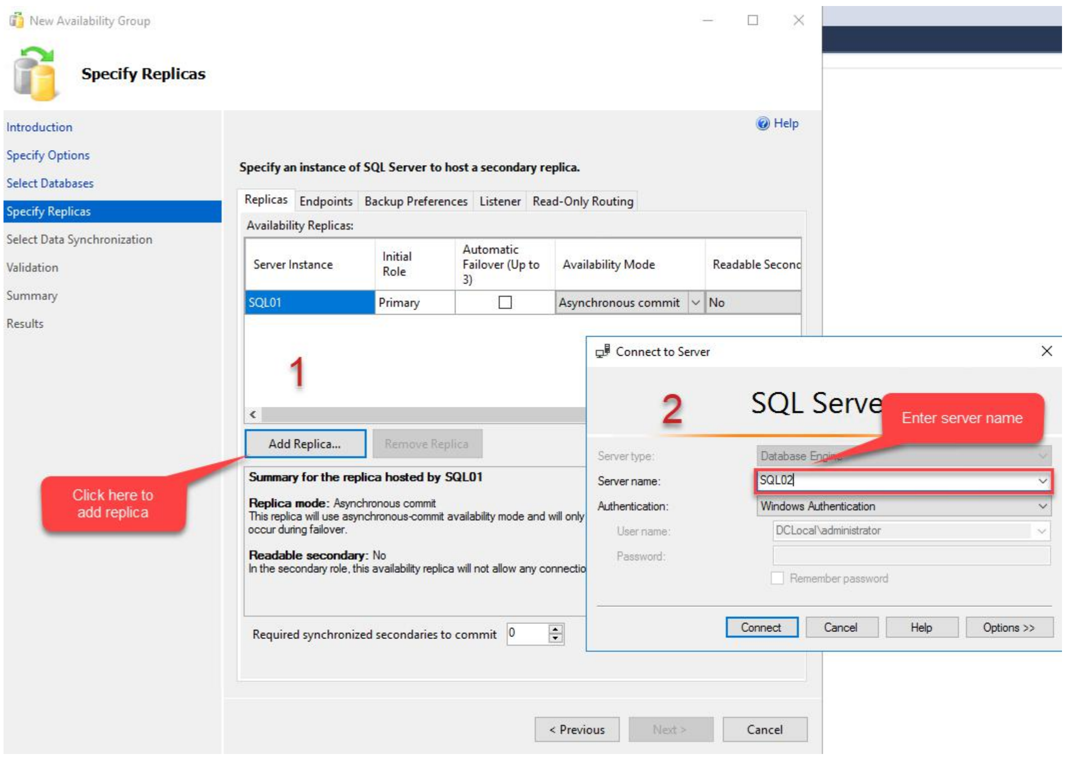
On the **Specify replica** screen, we will configure the following options:

1. List of availability replicas.
2. Endpoints.

3. Backup preference.
4. Availability Group Listener.
5. Read-Only routing.

Let me explain all the options.

First, in the **replica** tab, you can specify the list of replicas that you want to include in the availability group. We are going to include **SQL02.Dc.Local** and **SQL03.DC.Local** as the secondary replicas. To add a replica, click on the “**Add Replica**” button.


New Availability Group

Specify Replicas

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

Availability Replicas:

Server Instance	Initial Role	Automatic Failover (Up to 3)	Availability Mode	Readable Second
SQL01	Primary	☐	Asynchronous commit	No

Summary for the replica hosted by SQL01

Replica mode: Asynchronous commit
This replica will use asynchronous-commit availability mode and will only occur during failover.

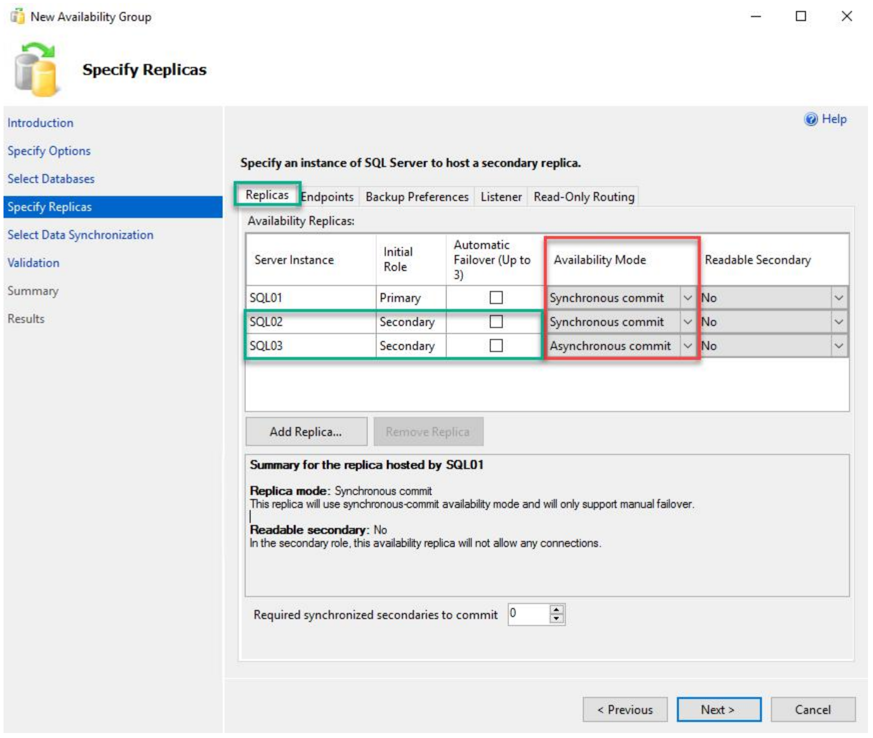
Readable secondary: No
In the secondary role, this availability replica will not allow any connectio

Required synchronized secondaries to commit: 0

Connect to Server

Server type: Database Engine
Server name: SQL02
Authentication: Windows Authentication
User name: DCLocal\administrator
Password:
☐ Remember password

When you click on “**Add replica**”, the **Connect to Server** dialog box opens. In the Server name text box, enter the name of the server that you want to add in the availability group, then click on **Connect**. Add **SQL02** in the server name text box and click on **Connect**. Similarly, add **SQL03.Dc.Local** in availability replicas. As I mentioned at the beginning of the article, **SQL02.Dc.Local** will be the synchronous replica and **SQL03.Dc.Local** will be the asynchronous replica; hence, choose Synchronous to commit from the Availability mode column mode for **SQL02.Dc.Local** and choose Asynchronous commit for **SQL03.Dc.Local**. See the following image:


New Availability Group

Specify Replicas

Introduction
Specify Options
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 | Read-Only Routing

Availability Replicas:

Server Instance	Initial Role	Automatic Failover (Up to 3)	Availability Mode	Readable Secondary
SQL01	Primary	☐	Synchronous commit	No
SQL02	Secondary	☐	Synchronous commit	No
SQL03	Secondary	☐	Asynchronous commit	No

Add Replica... | Remove Replica

Summary for the replica hosted by SQL01

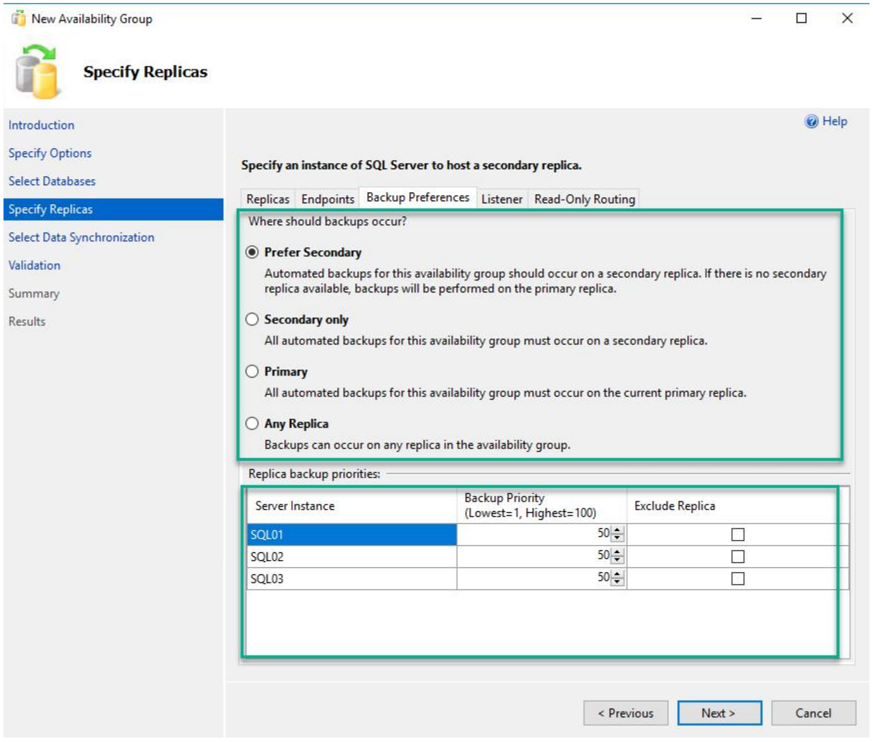
Replica mode: Synchronous commit
This replica will use synchronous-commit availability mode and will only support manual failover.

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

Required synchronized secondaries to commit 0

< Previous | Next > | Cancel

To configure the backup preference, click on the **Backup preference** tab. In the backup preference screen, you can see four options. The details of each option are provided on the screen, which is self-explanatory. As I mentioned, choose the "**Prefer secondary**" option. Backup priority will come up when you have configured the availability group with multiple secondary replicas, and the backup preference is secondary. Backup priority will be determined based on the number entered in the backup priority textbox. For example, if the backup priority is **70** for the **SQL03.Dc.Local** replica, the backup will be generated on the **SQL03.Dc.Local** replica. If you don't want to generate a backup on any specific replica, you can exclude a replica by clicking on the "**Exclude Replica**" checkbox. For now, do not make any changes to the backup priority. See the following image:


New Availability Group

Specify Replicas

Introduction
Specify Options
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 | Read-Only Routing

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.

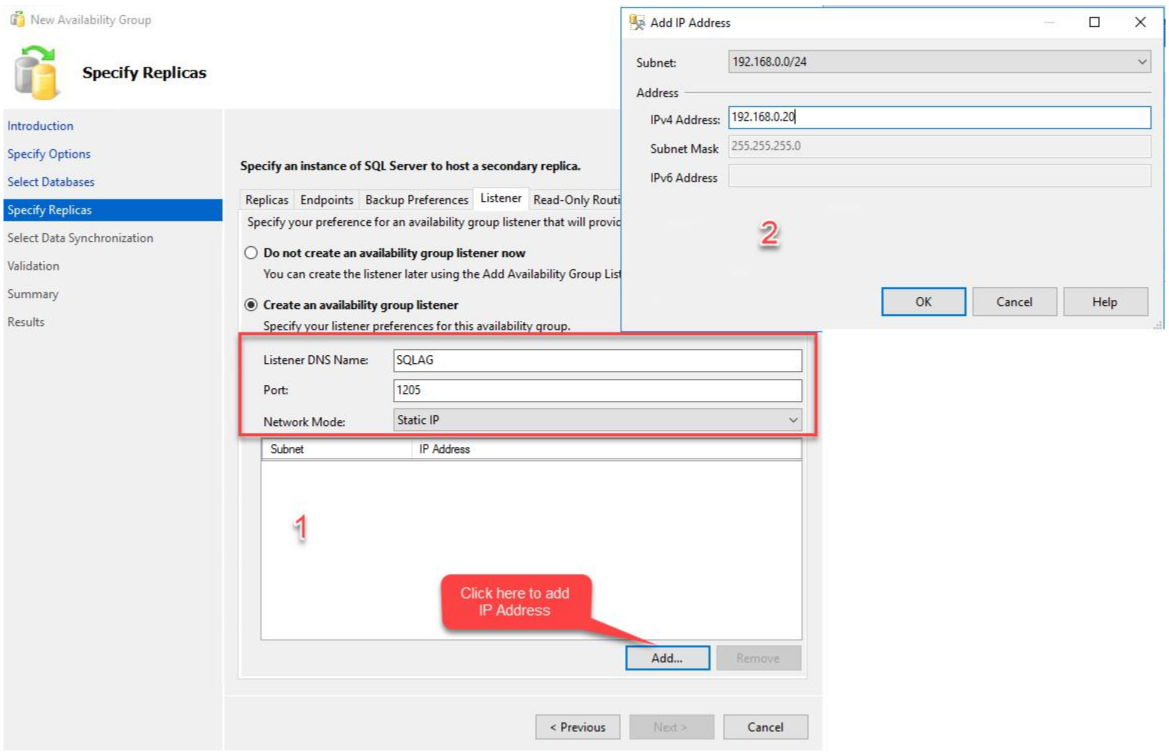
○ **Any Replica**
Backups can occur on any replica in the availability group.

Replica backup priorities:

Server Instance	Backup Priority (Lowest=1, Highest=100)	Exclude Replica
SQL01	50	☐
SQL02	50	☐
SQL03	50	☐

< Previous | Next > | Cancel

To create an availability group listener, click on the Listener tab. On the listener tab, select "**create new availability listener.**" In the DNS Name text box, provide the desired DNS name. This DNS name will be used to connect to the availability group. Enter the desired port in the Port number dialog box. Make sure the port is open in the windows firewall. Choose **Static IP Address** from the **Network** mode drop-down box. Click on the **Add button** to add IP Address. When you click on Add, the "**Add IP Address**" dialog box opens. In the dialog box, enter the desired IP Address. Click OK to close the dialog box. See the following image:



Listener DNS Name: SQLAG
Port: 1205
Network Mode: Static IP

Subnet: 192.168.0.0/24
IPv4 Address: 192.168.0.20
Subnet Mask: 255.255.255.0

New Availability Group
Specify Replicas
Introduction
Specify Options
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
Read-Only Routi
Specify your preference for an availability group listener that will provic
Do not create an availability group listener now
You can create the listener later using the Add Availability Group List
Create an availability group listener
Specify your listener preferences for this availability group.
Subnet
IP Address
1
Click here to add IP Address
Add...
Remove
< Previous
Next >
Cancel
Add IP Address
Address
IPv6 Address
2
OK
Cancel
Help

Once all parameters are configured, click on **Next**. On the "**Select initial data synchronization**" screen, you can see multiple options. They look the following way:

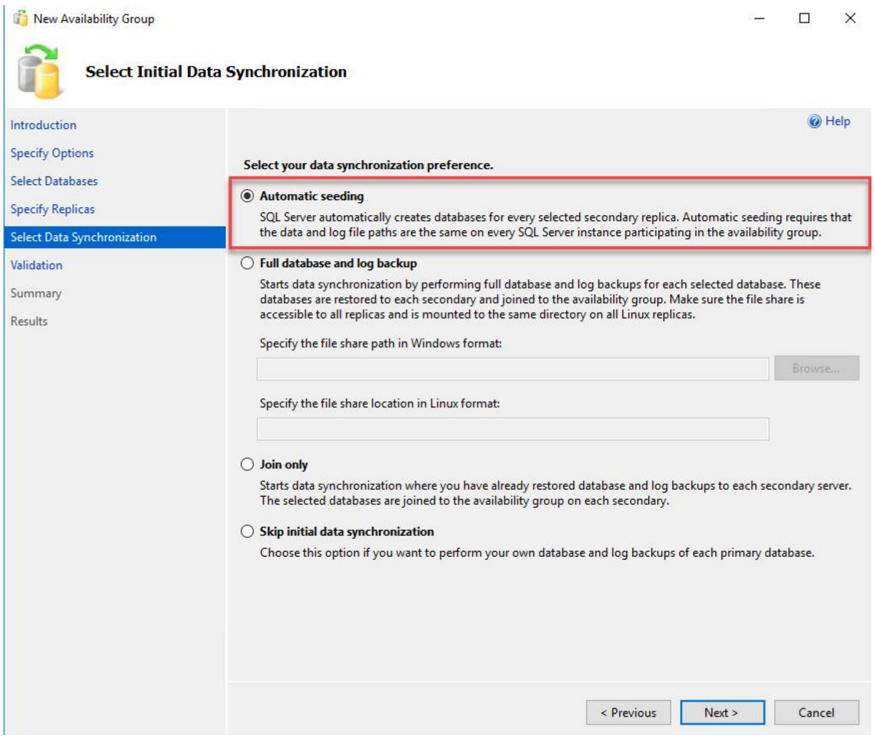
Automatic Seeding: When you choose this option, the wizard will automatically create an availability database on all secondary replicas. To use this option, we must make sure the data file and the log file paths are the same across all primary and secondary replicas.

Full Database and Log Backup: When you choose this option, Wizard will restore full database and log backups for available databases from shared locations entered in the "File Share Path" text box. To use this option, we must create a shared folder to keep full database and log backups. Your SQL Server service account must have read-write permission on the shared folder.

Join Only: When you choose this option, the wizard will join an availability database created on all secondary replicas. To use this option, we must restore a backup of the availability database on all secondary replicas.

Skip initial synchronization: When you choose this option, the wizard will skip the initial synchronization of primary and secondary replicas. We can perform it manually.

In our demo setup, data file and log file locations are the same; hence, choose the "Automatic Seeding" option from the "**data synchronization preference**" menu, then click on **Next**. See the following image:


New Availability Group

Select Initial Data Synchronization

Select your data synchronization preference.

(●) **Automatic seeding**
SQL Server automatically creates databases for every selected secondary replica. Automatic seeding requires that the data and log file paths are the same on every SQL Server instance participating in the availability group.

( ) **Full database and log backup**
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. Make sure the file share is accessible to all replicas and is mounted to the same directory on all Linux replicas.

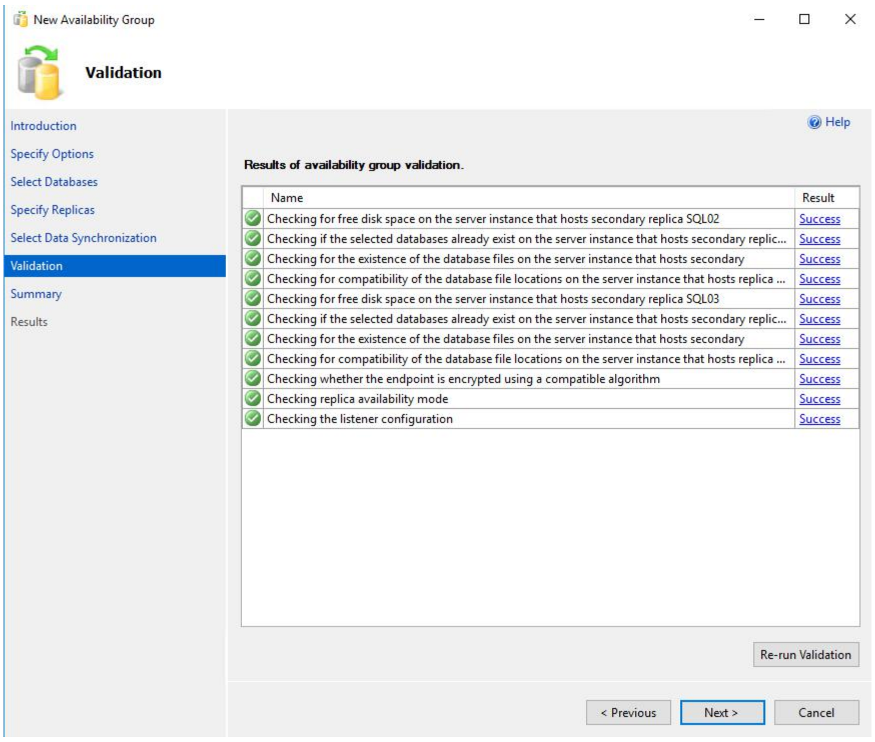
Specify the file share path in Windows format:

Specify the file share location in Linux format:

( ) **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.

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

On the validation screen, the wizard will perform validation checks on the entire configuration. It must be passed successfully. If you face any error during the validation test, you must fix it and click on the "**re-run the validation**" button to re-validate the configuration. Once the validation test passes successfully, click on **Next**. See the following image:

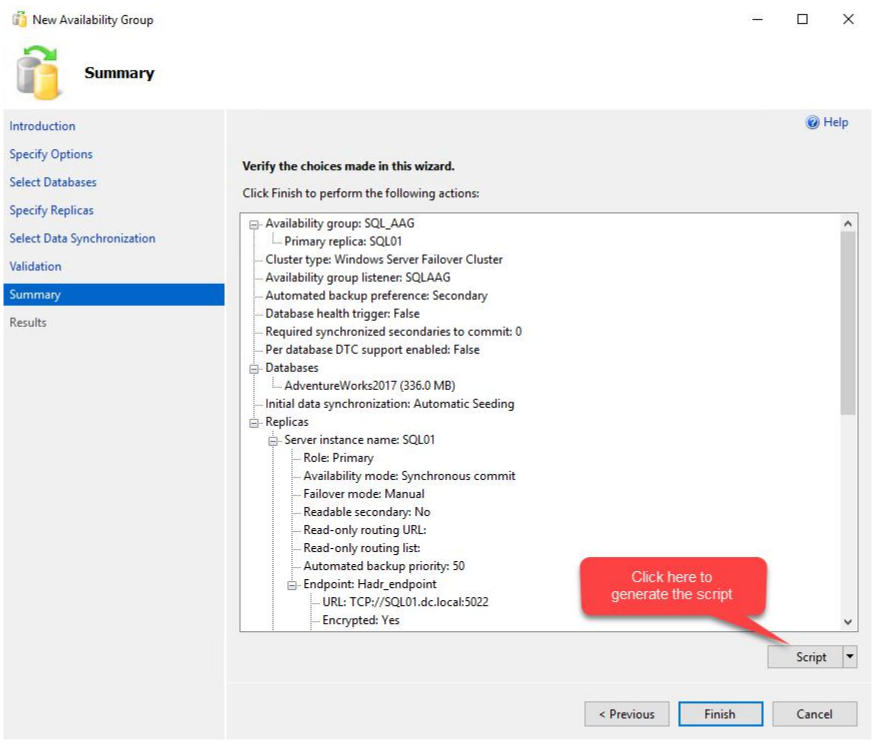

New Availability Group

Validation

Results of availability group validation.

	Name	Result
✓	Checking for free disk space on the server instance that hosts secondary replica SQL02	Success
✓	Checking if the selected databases already exist on the server instance that hosts secondary replic...	Success
✓	Checking for the existence of the database files on the server instance that hosts secondary	Success
✓	Checking for compatibility of the database file locations on the server instance that hosts replica ...	Success
✓	Checking for free disk space on the server instance that hosts secondary replica SQL03	Success
✓	Checking if the selected databases already exist on the server instance that hosts secondary replic...	Success
✓	Checking for the existence of the database files on the server instance that hosts secondary	Success
✓	Checking for compatibility of the database file locations on the server instance that hosts replica ...	Success
✓	Checking whether the endpoint is encrypted using a compatible algorithm	Success
✓	Checking replica availability mode	Success
✓	Checking the listener configuration	Success

On the **Summary** screen, review the whole configuration of parameters and click on the **Finish** button. Once you click on the finish button, the wizard will start the process of creating an availability group. You can also generate a script for it by clicking on the "Script" button. See the following image:


New Availability Group

Summary

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

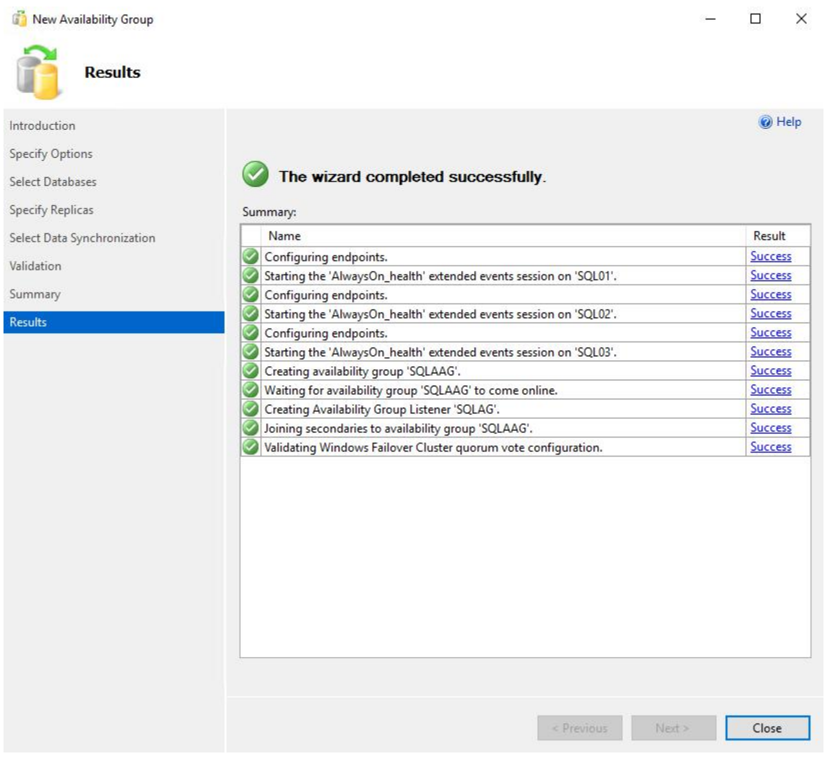
Help

Verify the choices made in this wizard.

Click Finish to perform the following actions:

```
Availability group: SQL_AAG
    Primary replica: SQL01
Cluster type: Windows Server Failover Cluster
Availability group listener: SQLAAG
Automated backup preference: Secondary
Database health trigger: False
Required synchronized secondaries to commit: 0
Per database DTC support enabled: False
Databases
    AdventureWorks2017 (336.0 MB)
Initial data synchronization: Automatic Seeding
Replicas
    Server instance name: SQL01
        Role: Primary
        Availability mode: Synchronous commit
        Failover mode: Manual
        Readable secondary: No
        Read-only routing URL:
        Read-only routing list:
        Automated backup priority: 50
        Endpoint: Hadr_endpoint
            URL: TCP://SQL01.dc.local:5022
            Encrypted: Yes
```

Congratulations. We have successfully created the Always On availability group. See the following image:


New Availability Group

Results

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

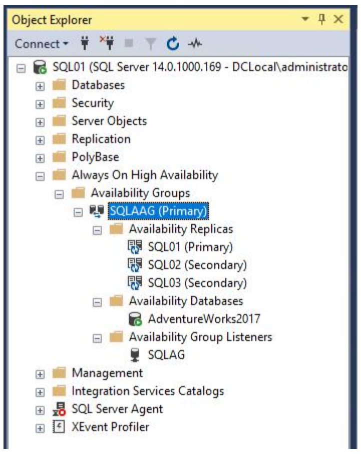
Help

The wizard completed successfully.

Summary:

Name	Result
Configuring endpoints.	Success
Starting the 'AlwaysOn_health' extended events session on 'SQL01'.	Success
Configuring endpoints.	Success
Starting the 'AlwaysOn_health' extended events session on 'SQL02'.	Success
Configuring endpoints.	Success
Starting the 'AlwaysOn_health' extended events session on 'SQL03'.	Success
Creating availability group 'SQLAAG'.	Success
Waiting for availability group 'SQLAAG' to come online.	Success
Creating Availability Group Listener 'SQLAG'.	Success
Joining secondaries to availability group 'SQLAAG'.	Success
Validating Windows Failover Cluster quorum vote configuration.	Success

To view details of the availability group, expand AlwaysOn High Availability in **object explorer** ? Expand **availability group**. You can see the availability group named “SQL_AAG” has been created. To view the participating node, expand **SQL_AAG** ? Expand **availability replica**. To view Availability databases, expand **Availability databases**. And to view Listener, expand **Availability group listener**. See the following image:


Object Explorer
Connect
SQL01 (SQL Server 14.0.1000.169 - DCLocal\administrato
Databases
Security
Server Objects
Replication
PolyBase
Always On High Availability
Availability Groups
SQLAAG (Primary)
Availability Replicas
SQL01 (Primary)
SQL02 (Secondary)
SQL03 (Secondary)
Availability Databases
AdventureWorks2017
Availability Group Listeners
SQLAG
Management
Integration Services Catalogs
SQL Server Agent
XEvent Profiler

Configure Read-only routing list in always on availability groups

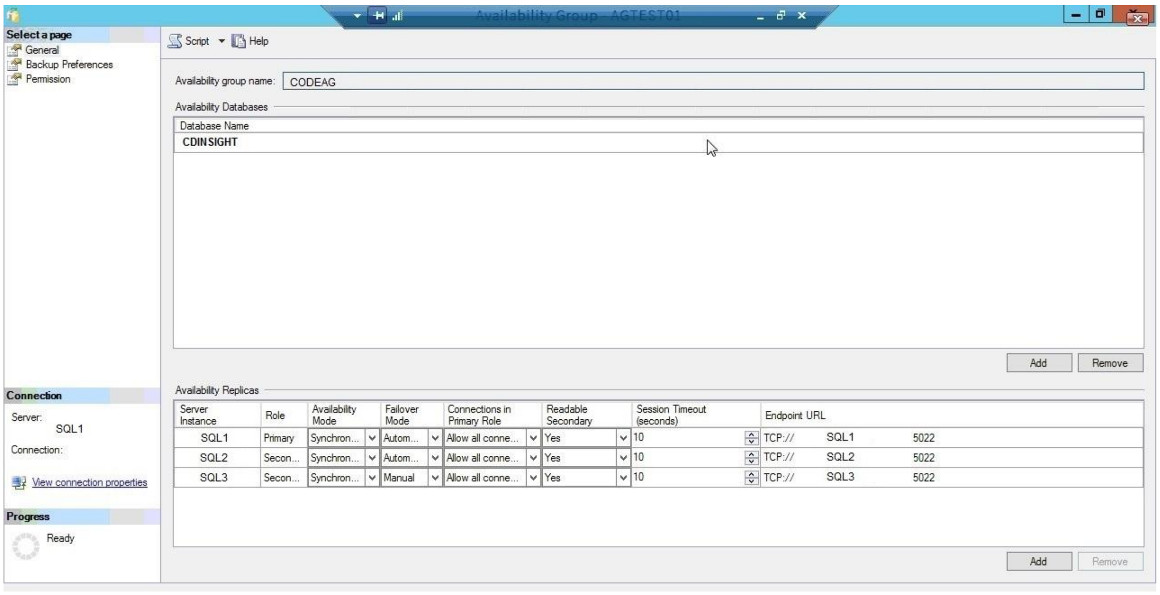
Read-only Routing for an Always On

As DBAs, we generally come across our clients complaining that the current Production Server is not able to hold the load on the server and whether the load may be balanced with the Secondary Server. This is possible with a database in DR Server with Read-only database in Log Shipping and Secondary SQL Server replicas in Always On Availability Group. The biggest advantage of Always On Groups is that it allows us to set up group level HA for any number of databases and we can create up to four secondary replicas and this is a combination of Clustering, Log Shipping and Database Mirroring where the data transmission is more flexible and functional.

Always On Readable Secondary Replica has a feature to handle particular read-only connection requests called Read-only routing. Generally, by default, both read and read-intent are directed to the Primary replica and nothing is intended for the secondary replicas. Now, the secondary replicas can be not just used for Backup, DBCC and reporting purposes but also can be queried in the future by using 'ReadOnly' as their Application Intent in the application connection string.

We have three replicas SQL1, SQL2 and SQL3 in the Windows failover cluster. Each node has a standalone SQL Server 2017 instance installed and configured with Always On AG. We are always on AG Group named "CODEAG" with the listener name "CODELIS"

In the following screenshot, SQL1 is a primary replica, SQL2 is a secondary replica, SQL3 is a secondary replica.



Availability group name: CODEAG

Availability Databases

Database Name
CDINSIGHT

Server: SQL1

Connection:

Availability Replicas

Server Instance	Role	Availability Mode	Failover Mode	Connections in Primary Role	Readable Secondary	Session Timeout (seconds)	Endpoint URL
SQL1	Primary	Synchron...	Autom...	Allow all conne...	Yes	10	TCP:// SQL1 5022
SQL2	Secon...	Synchron...	Autom...	Allow all conne...	Yes	10	TCP:// SQL2 5022
SQL3	Secon...	Synchron...	Manual	Allow all conne...	Yes	10	TCP:// SQL3 5022

Configure

Step1:

Connections are made to the Availability Group using the Listener name or IP. Now, to create multiple listeners for one availability group, please follow the below procedure.

To manually create or configure a listener for an availability group

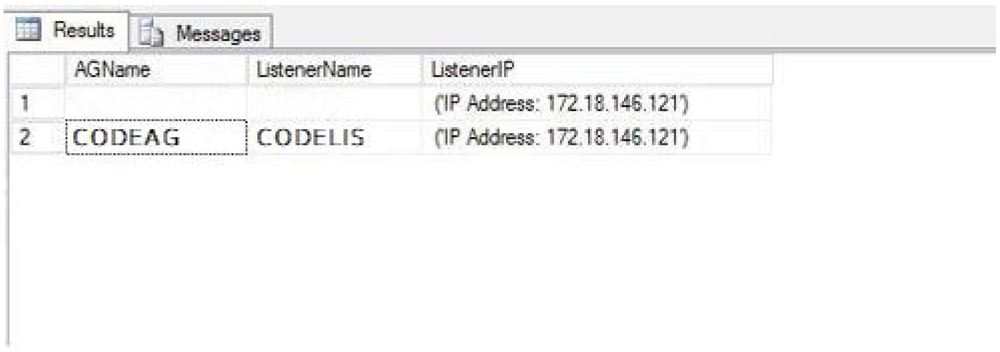
1. Under Object Explorer, connect to the instance that holds the primary replica of the availability group.
2. Expand the AON Group and click the Availability Group for which we are trying to manually configure the listener and proceed.
3. Right-click the Availability Group Listeners Node and select New Listener Command. This opens a new dialog box for configuring a listener.
4. Port Number of an existing listener can be changed by expanding the Availability Group Listeners Node followed by right-clicking the listeners and selecting the properties.

5. Now, enter the new Port Number and click OK.

Identify the listener name configured for Always On replication by querying DMV as below.

```
SELECT AV.name AS AVGName
, AVGLis.dns_name AS ListenerName
, AVGLis.ip_configuration_string_from_cluster AS ListenerIP
FROM sys.availability_group_listeners AVGLis
INNER JOIN sys.availability_groups AV on AV.group_id = AV.group_id
```

In the following screenshot, the AG group name is CODEAG and AG listener IP is CODELIS.



	AGName	ListenerName	ListenerIP
1			('IP Address: 172.18.146.121')
2	CODEAG	CODELIS	('IP Address: 172.18.146.121')

Step 2:

To configure read-only routing, we need to configure the replicas to read-intent only to allow read-only connections to secondary replicas.

1. Connect to the instance that holds the Primary replica.
2. Expand the AON High Availability Node, then AG Group Node.
3. Click the AG Group whose replica should be changed.
4. Right-click the replica and select properties to change the connection access for the Primary and Secondary roles as follows.

The Secondary role has a new value from the readable secondary drop list.

Read-Intent only

This option allows reading access of the secondary databases of this replica. Only read-only connections are allowed.

Yes

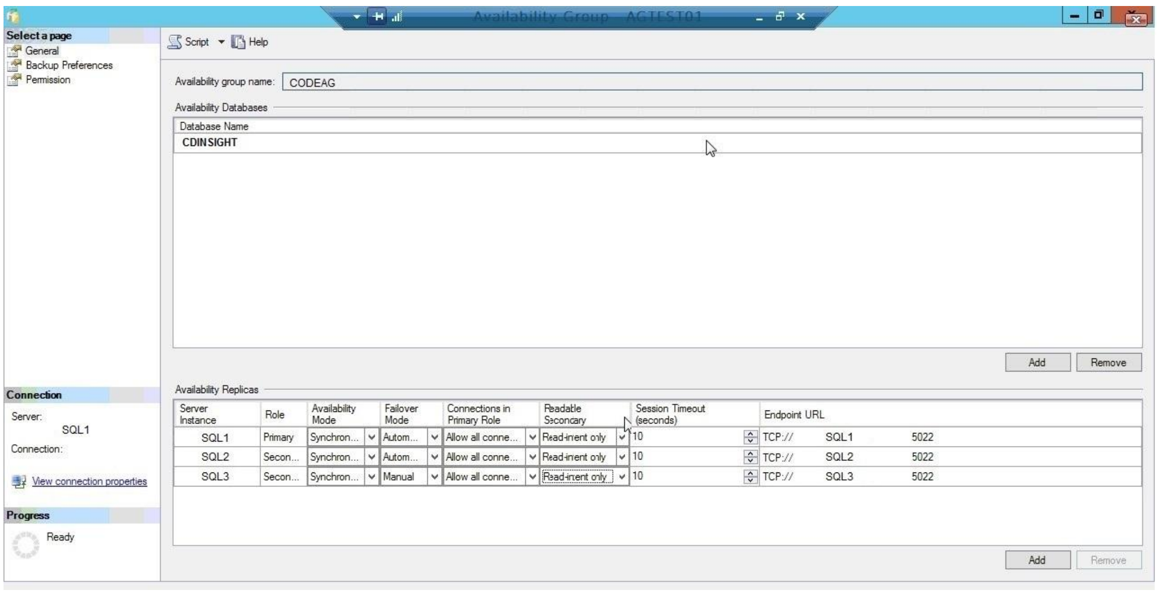
This option allows to read access only but all the connections are allowed for the secondary replica.

No

This stops all the user connections to the secondary replica and doesn't even allow you to read.

Set the readable secondary properties to **Read-intent only.**

In the following screenshot, the Readable Secondary properties of each secondary replica are set to Read-intent only.



Select a page
- General
- Backup Preferences
- Permission

Availability group name: CODEAG

Availability Databases

Database Name
CDINSIGHT

Availability Replicas

Server Instance	Role	Availability Mode	Failover Mode	Connections in Primary Role	Readable Secondary	Session Timeout (seconds)	Endpoint URL
SQL1	Primary	Synchron...	Autom...	Allow all conne...	Read-intent only	10	TCP:// SQL1 5022
SQL2	Secon...	Synchron...	Autom...	Allow all conne...	Read-intent only	10	TCP:// SQL2 5022
SQL3	Secon...	Synchron...	Manual	Allow all conne...	Read-intent only	10	TCP:// SQL3 5022

Connection
Server: SQL1
Connection:

Progress
Ready

Step 3:

Each readable secondary replica can be assigned a read-only routing URL that will be used for routing read-intent connection requests to a specific readable secondary replica.

Use T-SQL to specify a read-only routing URL for all of the replicas in our Availability Group.

ALTER AVAILABILITY GROUP [CODEAG]

MODIFY REPLICA ON

N'SQL1' WITH

(SECONDARY_ROLE (ALLOW_CONNECTIONS = READ_ONLY));

ALTER AVAILABILITY GROUP [CODEAG]

MODIFY REPLICA ON

N'SQL1' WITH

(SECONDARY_ROLE (READ_ONLY_ROUTING_URL = N'TCP://SQL1.abc.com:17999'));

ALTER AVAILABILITY GROUP [CODEAG]

MODIFY REPLICA ON

N'SQL2' WITH

(SECONDARY_ROLE (ALLOW_CONNECTIONS = READ_ONLY));

ALTER AVAILABILITY GROUP [CODEAG]

MODIFY REPLICA ON

N'SQL2' WITH

(SECONDARY_ROLE (READ_ONLY_ROUTING_URL = N'TCP://SQL2.abc.com:17999'));

```
ALTER AVAILABILITY GROUP [CODEAG]
MODIFY REPLICA ON
N'SQL3' WITH
(SECONDARY_ROLE (ALLOW_CONNECTIONS = READ_ONLY));

ALTER AVAILABILITY GROUP [CODEAG]
MODIFY REPLICA ON
N'SQL3' WITH
(SECONDARY_ROLE (READ_ONLY_ROUTING_URL = N'TCP://SQL3.abc.com:17999'));
```

Step 4:

For the replica that we are marking as read-only routing when it is the primary replica, there is a need to specify a read-only routing list and this gets effected only when the local replica is running under the primary role.

```
ALTER AVAILABILITY GROUP [CODEAG]
MODIFY REPLICA ON
N'SQL1' WITH
(PRIMARY_ROLE (READ_ONLY_ROUTING_LIST=('SQL2','SQL3')));

ALTER AVAILABILITY GROUP [CODEAG]
MODIFY REPLICA ON
N'SQL3' WITH
(PRIMARY_ROLE (READ_ONLY_ROUTING_LIST=('SQL2','SQL1')));

ALTER AVAILABILITY GROUP [CODEAG]
MODIFY REPLICA ON
N'SQL2' WITH
(PRIMARY_ROLE (READ_ONLY_ROUTING_LIST=('SQL3','SQL1')));
```

In the above script, the example when SQL1 is the primary replica, the read-intent only workload will be redirected to the readable secondary replicas; the SQL2 and SQL3 respectively: similarly, if SQL3 is the primary replica, the read-intent only workload will be redirected to the readable secondary replicas; the SQL2 and SQL1 respectively.

Read-only directed traffic will be routed to the first available replica until and unless it is not accessible it would direct the traffic to the next available replica in the routing list. When you have more than one replica, it is not possible to share the load between replicas till SQL Server 2012 and 2014. But, SQL server 2016 allows you to balance the load between read-only replicas.

Example:

```
ALTER AVAILABILITY GROUP [CODEAG]
MODIFY REPLICA ON
N'SQL2' WITH
(PRIMARY_ROLE (READ_ONLY_ROUTING_LIST=(('SQL3','SQL1'), ‘SQL2’)));
```

This routing list behavior ‘load balances’ read-only connections between SQL3 and SQL1. Above, we have two embedded lists in the routing list:

List 1: ‘SQL3’, ‘SQL1’

List 2: ‘SQL2’

Route to the replicas in the first list. SQL3 and SQL1 are accessible to read-only connections. The first incoming read-only connection will be routed to SQL3, the second read-only connection will be routed to SQL1, the third read-only connection will be routed to SQL3, the fourth read-only connection will be routed to SQL1, and so on, with a ‘round-robin’ distribution of read-only connections between the two replicas in the first list.

If any replicas become unavailable, routing will continue with remaining replicas in the first list. If SQL3 or SQL1 becomes inaccessible to read-only connections, then the read-only connections will only be routed to the accessible read-only replicas in the first list. For example, if SQL3 is in a not synchronized state or ALLOW_CONNECTIONS is set to NO, then all read-only connections will be routed to SQL1. So long as one of the servers is available for read-only connections, NO read-only connections will be routed to SQL2.

If all replicas in the first list are inaccessible, route to replicas in next list. If SQL3 and SQL1 become inaccessible to read-only connections, then all read-only connections will only be routed to the next list of replicas, which in this case is SQL2.

Resume routing to the first list if any replicas become available. As secondary replicas that have higher priority in the list become accessible to read-only connections, future read-only connections will connect to them as appropriate.

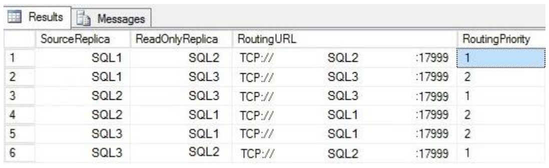
In SQL Server 2016, you can configure load-balancing across a set of read-only replicas.

Step 5:

sys.availability_read_only_routing_lists DMV, that returns the read-only routing list of each Availability Group replica in the Always On Availability Group.

```
SELECT      AVGSrc.replica_server_name AS SourceReplica
, AVGRepl.replica_server_name AS ReadOnlyReplica
, AVGRepl.read_only_routing_url AS RoutingURL
, AVGRL.routing_priority AS RoutingPriority
FROM sys.availability_read_only_routing_lists AVGRL
INNER JOIN sys.availability_replicas AVGSrc ON AVGRL.replica_id = AVGSrc.replica_id
INNER JOIN sys.availability_replicas AVGRepl ON AVGRL.read_only_replica_id = AVGRepl.replica_id
INNER JOIN sys.availability_groups AV ON AV.group_id = AVGSrc.group_id
ORDER BY SourceReplica
```

In the following screenshot, the result shows routing URL and routing priority.



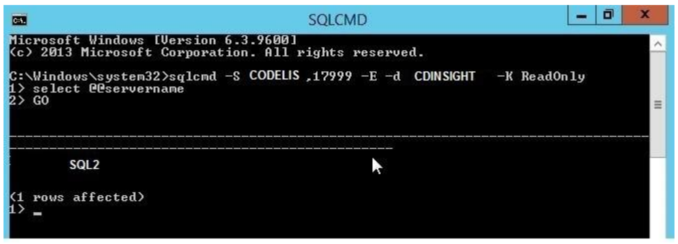
Results | Messages

	SourceReplica	ReadOnlyReplica	RoutingURL			RoutingPriority
1	SQL1	SQL2	TCP://	SQL2	:17999	1
2	SQL1	SQL3	TCP://	SQL3	:17999	2
3	SQL2	SQL3	TCP://	SQL3	:17999	1
4	SQL2	SQL1	TCP://	SQL1	:17999	2
5	SQL3	SQL1	TCP://	SQL1	:17999	2
6	SQL3	SQL2	TCP://	SQL2	:17999	1

Step 6:

To test read-only routing using SQLCMD, use –K ReadOnly parameter that shows secondary replica receiving read connections as per the routing list.

In the following screenshot, the secondary replica is accepting the read connections i.e… SQL2.



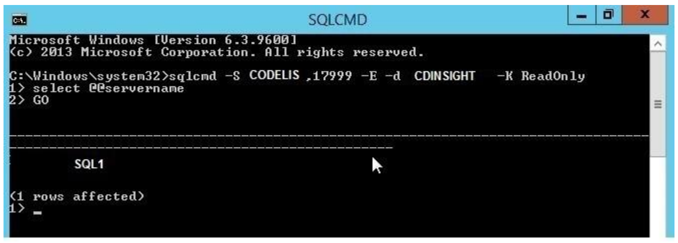
```
SQLCMD
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Windows\system32>sqlcmd -S CODELIS ,17999 -E -d CDINSIGHT   -K ReadOnly
1> select @@servername
2> GO

--------------------------------------------------------------------------------
------------------------------------------------

        SQL2

(1 rows affected)
1> _
```

Step 7:

Failover the availability group and test read-only routing.

1. In Object Explorer, connect to a server instance that hosts a secondary replica of the availability group that needs to be failed over. Expand the server tree.
2. Expand the AlwaysOn High Availability node and the Availability Groups node.
3. Right-click the availability group to be failed over, and select Failover.

Now, the SQL2 becomes primary replica and connections are handled by SQL1.



```
SQLCMD
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Windows\system32>sqlcmd -S CODELIS ,17999 -E -d CDINSIGHT   -K ReadOnly
1> select @@servername
2> GO

--------------------------------------------------------------------------------
------------------------------------------------

        SQL1

(1 rows affected)
1> _
```

The Connection String Syntax that application should use will depend on the SQL Server Provider.

If it is .Net Framework Data Provider 4.0.2 for SQL Server, the syntax will be as follows:

Server=tcp:MyAgListener,portnumber;Database=SQL1;IntegratedSecurity=SSPI;ApplicationIntent=ReadOnly;MultiSubnetFailover=True

To bring down the workloads, this Read-only routing remains the best option. An application involving SQL Server Reporting Services hosted in SharePoint or the Native Mode installation of Report Server can use Read-only intent that avoids typical blocking, Memory and CPU Usage on Primary Nodes.

2) Availability Groups Implementation without Failover Clustering (with readable secondary - Read Scale Availability Group new in SQL Server 2017)

It can be used for report server where no need to create failover cluster hence no High Availability and No Disaster Recovery only use secondary replica as read only report server to offload workload.

The setup is the same that you'd do for a regular availability group, with two distinctions:

3. Windows Server Failover Clustering is not needed
    1. The Availability Group manager does not have or need a cluster context
4. A new cluster_type option is present

    1. To use this, we want the cluster_type of "NONE"

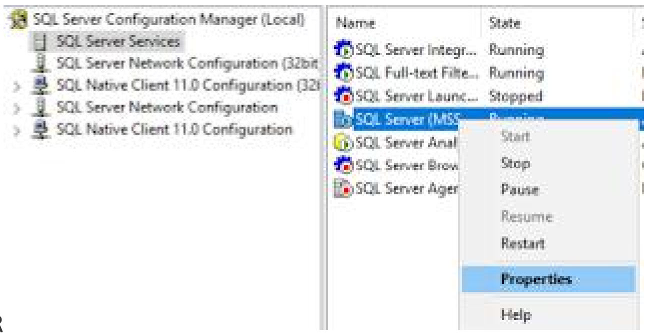
Follow Step 3 and Step 4 from Regular Availability Group.

Details Steps are given bellow:

Configure Always on AG with SQL 2017 (Cluster less AG)

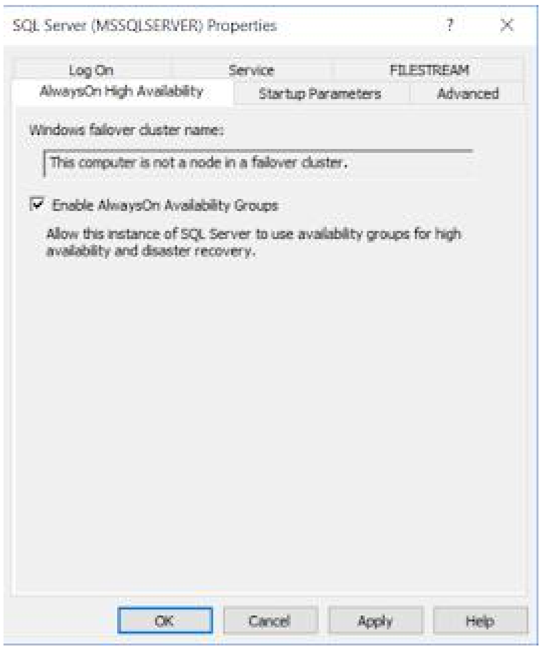
One of the new features that we have with SQL 2017 is that you no longer need a Windows cluster to enable the AlwaysOn feature with SQL server
In this post I'll be showing you how to configure an Always On availability group with SQL Server 2017, most of the steps are the same for the older versions and I'll be telling the differences on each of the steps so lets get started.

1. Enabling AlwaysOn: In order to do that, you need to go to SQL Server Configuration Manager, and on the SQL Server Services, hit properties over the


SQL Server Configuration Manager (Local)
SQL Server Services
SQL Server Network Configuration (32bit
SQL Native Client 11.0 Configuration (32
SQL Server Network Configuration
SQL Native Client 11.0 Configuration
Name
State
SQL Server Integr... Running
SQL Full-text Filte... Running
SQL Server Launc... Stopped
SQL Server (MSS
SQL Server Anal
SQL Server Brow
SQL Server Ager
Start
Stop
Pause
Resume
Restart
Properties
Help

MSSQLSERVER

2. In the properties window, look for "AlwaysOn High Availability", if you are running with an older version, the checkbox "Enable AlwaysOn Availability Groups" won't be availabile until you make the machine where SQL Server part of a cluster but just part of the windows cluster, the SQL Server instance remains as an Stand Alone type. Once that you have the checkbox enabled, check it, and click apply and Ok. Enabling this feature requires a Service restart so take that in account. Also, you need to do this in all the SQL instances that will be part of the Availability Group.



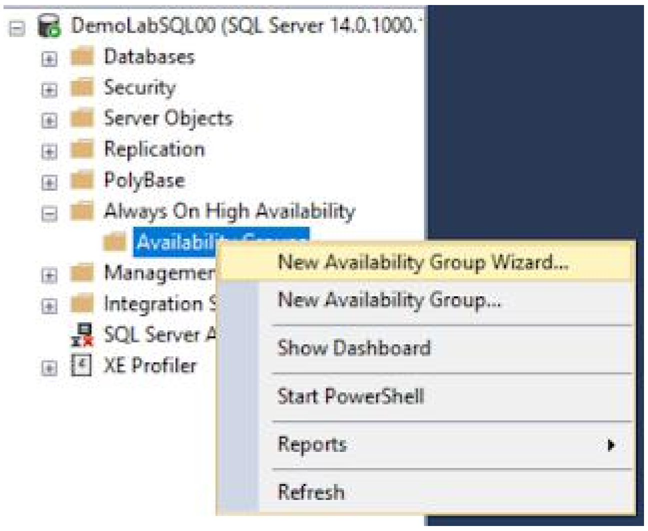
Windows failover cluster name:

This computer is not a node in a failover cluster.

☑ Enable AlwaysOn Availability Groups

Allow this instance of SQL Server to use availability groups for high availability and disaster recovery.

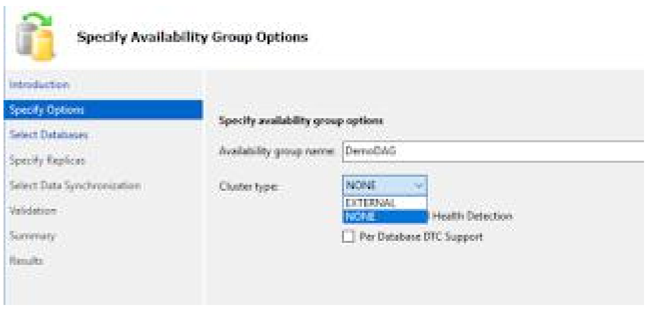
3. Go to SQL Management studio, connect to any of the SQL instances that will be part of the group, look for the "Always On High Availability" folder, expand it and you will see a folder with the name "Availability Groups" do a right click on it and select "New Availability Group Wizard…"



4. In the availability group wizard, configure the name you want for your availability group.

a. Cluster Type (New with SQL 2017): you have 2 options here (the official documentation shows 3), External or None, if you choose External, it means that you will joining this availability group to an external cluster, E.g. if you have a windows cluster on your primary datacenter that is on a different network or you want to add it to a Linux cluster.
b. Database health level detection (New with SQL 2017), this will enable a constant validation of the databases that are part of the availability group that if anything goes wrong with any of the databases it will trigger an automatic failover
c. Per Database DTC support (New with 2016 SP1) Allows the Distributed Transaction Coordinator through the availability group, feature that was not available in older versions of SQL (2012 and 2014)

If you don't have a cluster created and just like this scenario, choose **NONE**



Specify availability group options

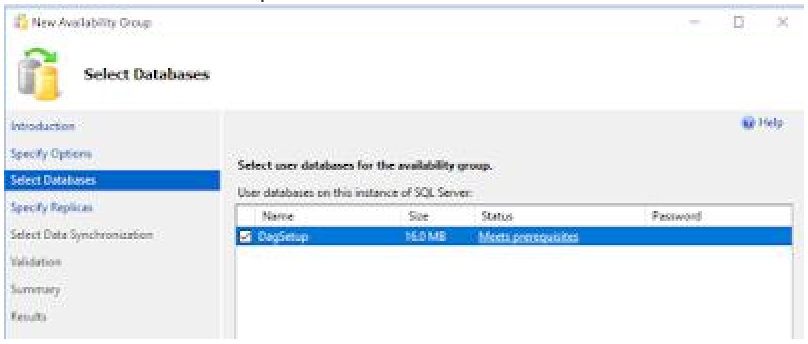
Availability group name: DemoDAG

Cluster type: NONE

Health Detection

Per Database DTC Support

5. Select the databases that will be part of the availability group. Requisites for a database to be considered:
   a. Be on Full recovery model
   b. Have a full backup



Select user databases for the availability group.

User databases on this instance of SQL Server:

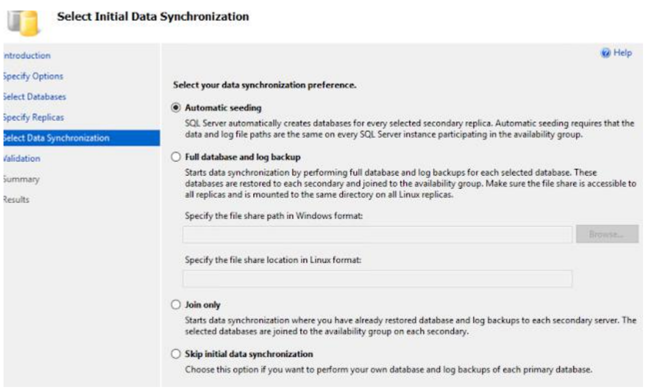
Name	Size	Status	Password
☑ DagSetup	16.0 MB	Meets prerequisites	

6. In here you will be choosing the replicas that you want to add to your group. Important features here:

1. Initial Role: You will set the role that each replica will have once that the group becomes available
2. Failover Mode: Manual or Automatic
3. Availability Mode: Synchronous commit or Asynchronous commit
4. Readable Secondary: No, Yes - Read Intent Only and Yes. If you choose No or Yes - Read Intent Only, you won't be available to query your replicas, the Read Intent is when you enable that feature that routes the read operations to your available replicas. With the Yes option you will be able to query the databases in your replica.
5. The endpoints tab will show you the URL and ports each replica is set to.
6. Backup preferences tab: in here you can configure if you want to take the backups from the primary or the secondary servers.
7. Listener tab: In this window it lets you set the parameters to configure the availability group listener, however my advice is to do it later, configure your group first and once that is done, configure it later.
8. Read-Only Routing: tab This lets you configure your read only routing for the read-intent setup, this allows you to load balance the queries so you have only the queries that will Insert, Delete or Update on the primary and all the Select queries routed to your secondary's so they don't consume resources the Primary will use (I'll explain it in another post)

7.

1. Automatic seeding (Starting SQL 2016): With this type, SQL Server will do everything for you, it will use the default folders configured so make sure you have enough space available on it.
2. Full database and log backup: Same as the Automatic seeding, however in this case it lets you choose where do you want to generate the backups
3. Join Only: With this one, you do prepare everything in your replica, important notice, you do a restore with norecovery in your replica and apply at least one log backup. This option is useful with really large databases.
4. Skip initial data synchronization: Same as before, however in this particular one, you need to restore everything just at the time before initializing the synchronization otherwise it will tell you that there are items pending to be restored.
5. Failover Mode: Manual or Automatic
6. Availability Mode: Synchronous commit or Asynchronous commit
7. Readable Secondary: No, Yes - Read Intent Only and Yes. If you choose No or Yes - Read Intent Only, you won't be available to query your replicas, the Read Intent is when you enable that feature that routes the read operations to your available replicas. With the Yes option you will be able to query the databases in your replica.
8. The endpoints tab will show you the URL and ports each replica is set to.
9. Backup preferences tab: in here you can configure if you want to take the backups from the primary or the secondary servers.
10. Listener tab: In this window it lets you set the parameters to configure the availability group listener, however my advice is to do it later, configure your group first and once that is done, configure it later.
11. Read-Only Routing: tab This lets you configure your read only routing for the read-intent setup, this allows you to load balance the queries so you have only the queries that will Insert, Delete or Update on the primary and all the Select queries routed to your secondary's so they don't consume resources the Primary will use (I'll explain it in another post)



Select your data synchronization preference.

[✓ Automatic seeding]
SQL Server automatically creates databases for every selected secondary replica. Automatic seeding requires that the data and log file paths are the same on every SQL Server instance participating in the availability group.

Full database and log backup
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. Make sure the file share is accessible to all replicas and is mounted to the same directory on all Linux replicas.

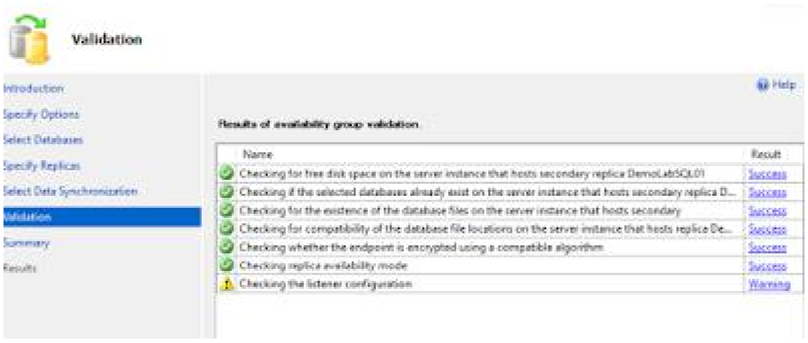
Specify the file share path in Windows format:

Specify the file share location in Linux format:

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.

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

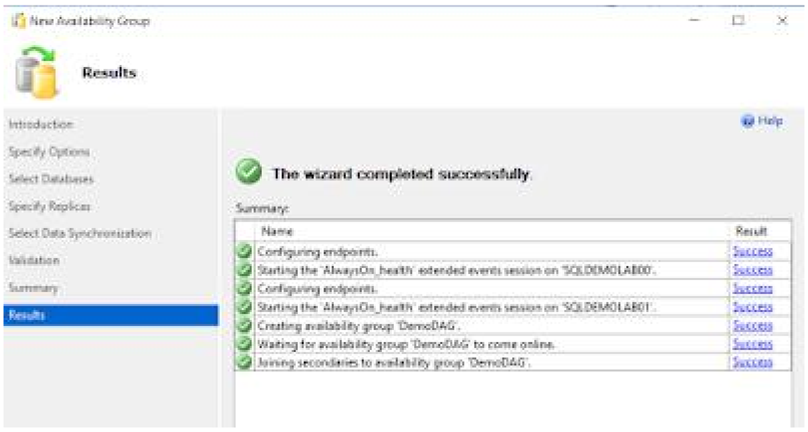
8. Verify that the validation runs successfully, this warning is because I didn't setup the listener in step 6, but that is fine, every time that I've tried to configure it from the very first page it fails, so my advice is that you configure it later (part of this post).



Results of availability group validation

	Name	Result
✓	Checking for free disk space on the server instance that hosts secondary replica DemoLabSQL01	Success
✓	Checking if the selected databases already exist on the server instance that hosts secondary replica D...	Success
✓	Checking for the existence of the database files on the server instance that hosts secondary	Success
✓	Checking for compatibility of the database file locations on the server instance that hosts replica De...	Success
✓	Checking whether the endpoint is encrypted using a compatible algorithm	Success
✓	Checking replica availability mode	Success
⚠	Checking the listener configuration	Warning

9. After hitting finish and if none of the steps have failed you will see the screen just like this.



The wizard completed successfully.

Summary:

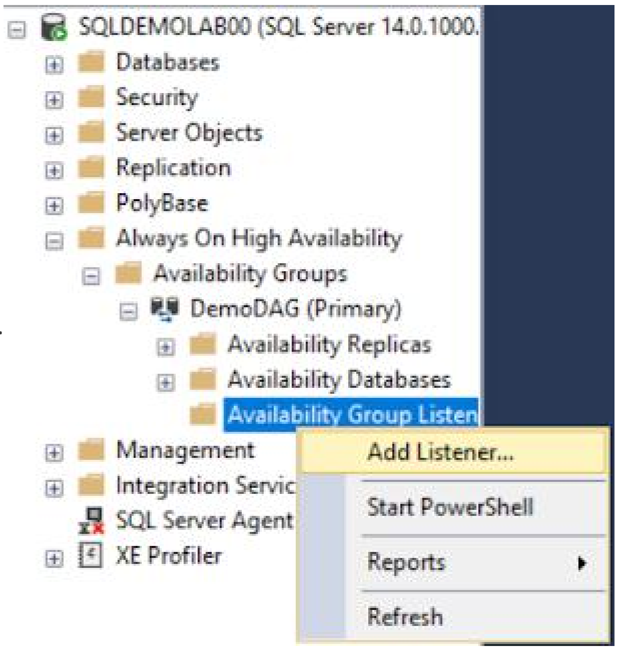
	Name	Result
✓	Configuring endpoints.	Success
✓	Starting the 'AlwaysOn_health' extended events session on 'SQLDEMOLAB00'.	Success
✓	Configuring endpoints.	Success
✓	Starting the 'AlwaysOn_health' extended events session on 'SQLDEMOLAB01'.	Success
✓	Creating availability group 'DemoDAG'.	Success
✓	Waiting for availability group 'DemoDAG' to come online.	Success
✓	Joining secondaries to availability group 'DemoDAG'.	Success

Setup of the listener

1.
The listener will be your single point of entry for your availability group, no matter which server is primary and which secondary, you will only need to configure your applications to use the Listener fqn or the ip and that will give you that High Availability you are looking with Always On.

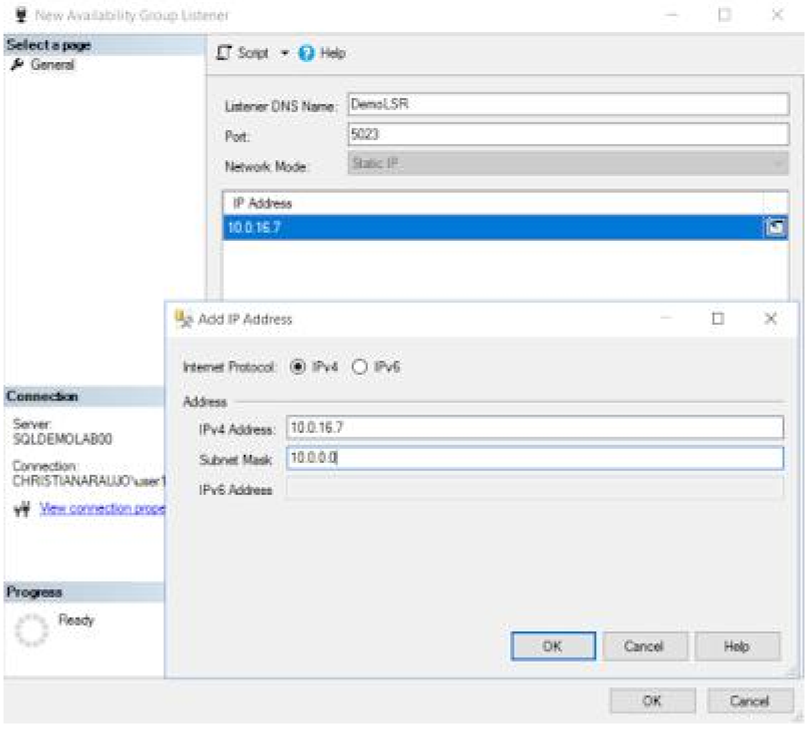
2.

1. Go to the "Always On High Availability" folder, expand your availability group and look for the "Availability group listener" and right click on



it.

1. In the Configuration screen, Configure the domain name you want your listener to respond onto, this will be like another computer in the domain, so be sure that you have permissions to create objects in the active directory, if you don't ask your domain admin to provision that name and grant you permissions over it so you can enable it.
    1. Configure the port you want the listener to listen to, this needs to be different than the endpoints and than the usual 1433 that the SQL instance will be listen to.
    2. Configure the IP: make sure that its an available IP in the domain, also make sure that you choose the ip from the same subnet that one of your replicas is running.


New Availability Group Listener

Listener DNS Name: DemoLSR
Port: 5023
Network Mode: Static IP

IP Address
10.0.16.7

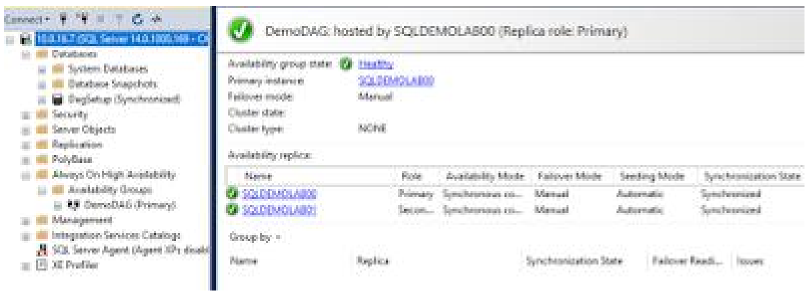
Connection
Server: SQLDEMOLAB00
Connection: CHRISTIANARAUJO\user1

Progress
Ready

Add IP Address

Internet Protocol: [✓ IPv4] IPv6

Address
IPv4 Address: 10.0.16.7
Subnet Mask: 10.0.0.0
IPv6 Address:



DemoDAG: hosted by SQLDEMOLAB00 (Replica role: Primary)

Availability group state:	✓ Healthy
Primary instance:	SQLDEMOLAB00
Failover mode:	Manual
Cluster state:	
Cluster type:	NONE

Availability replica:

	Name	Role	Availability Mode	Failover Mode	Seeding Mode	Synchronization State
✓	SQLDEMOLAB00	Primary	Synchronous co...	Manual	Automatic	Synchronized
✓	SQLDEMOLAB01	Secon...	Synchronous co...	Manual	Automatic	Synchronized

Group by ▾

Name	Replica	Synchronization State	Failover Readi...	Issues

You are all set, you can start configuring your applications to the listener and taking the advantages always On provides.