BAG (Basic Availability Group) Setup for MS SQL in AWS instances

Introduction, background and scope:

MS SQL natively provides BAG feature for DR(Disaster Recovery) and HA(High Availability). For a detailed workflow and insights we will be trying to install MS SQL in AWS instances and then perform the setup. The goal is to survive a server or data center failure with near-zero data loss, which requires low-latency, synchronous communication. Although for DR scenarios we should be doing for different regions, for a small scale HA setup we are trying the setup for the same region but different Availability Zones.

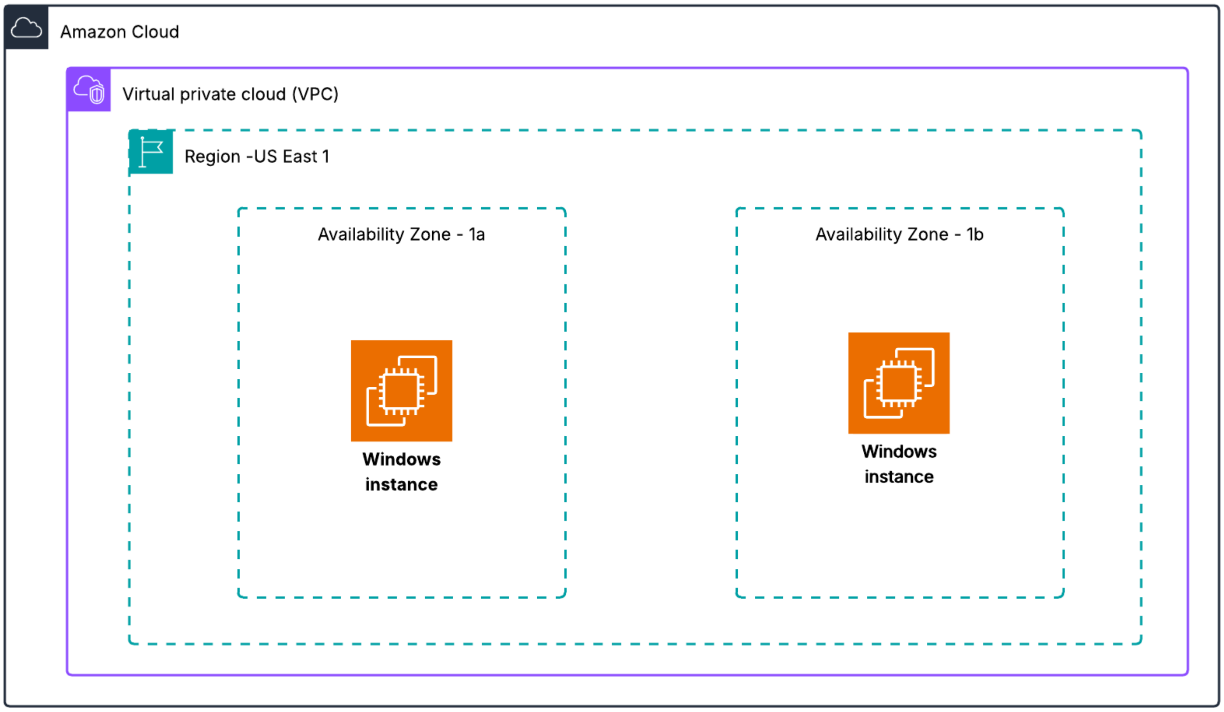


Figure: Simple Architecture Diagram for our setup

First, let's clarify the limitations of a Basic Availability Group (BAG) in SQL Server Standard Edition:

One Database per Group:
We can only have one user database in a BAG. If we need to protect multiple databases, you must create multiple BAGs.

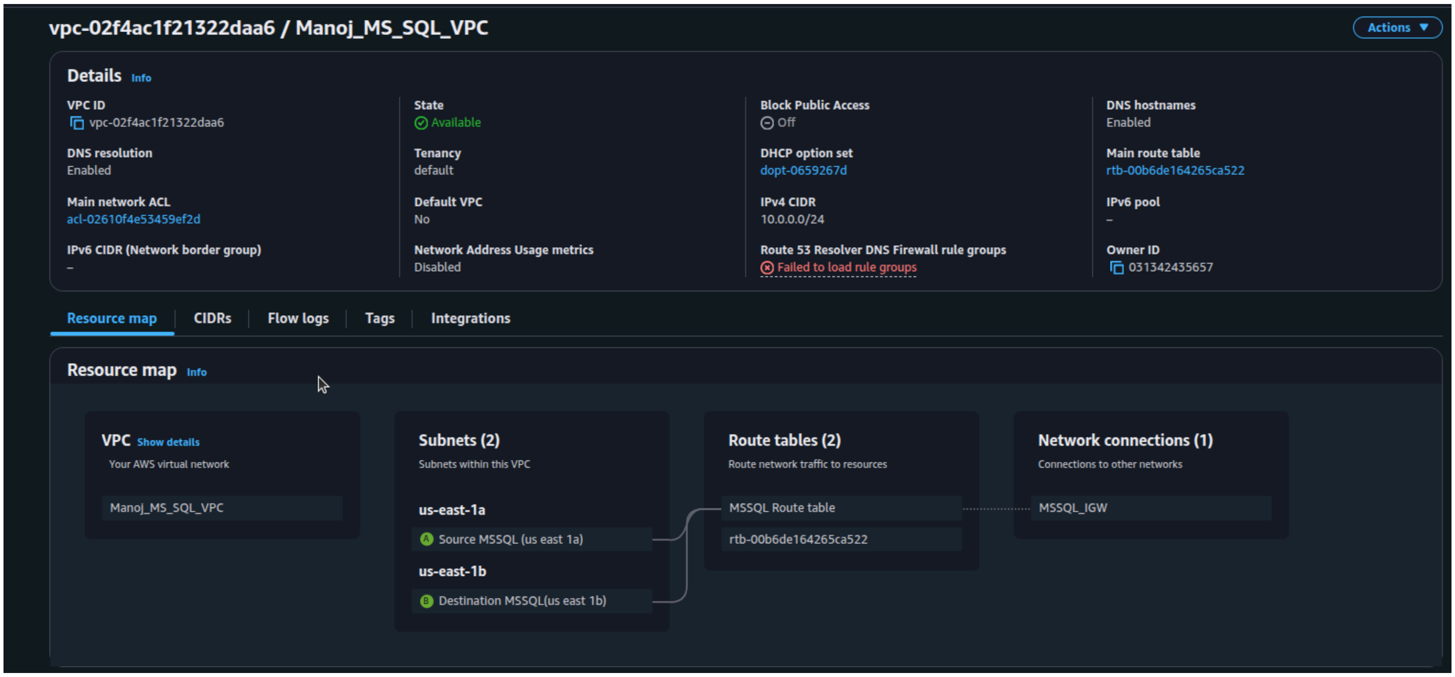
Two Replicas Only:
A BAG is limited to a primary replica and one secondary replica.

No Read Access on Secondary:
The secondary replica cannot be used for reading data or offloading backups. It is purely a hot standby.

Asynchronous-Commit Recommended:
While synchronous-commit is possible, it's generally recommended to use asynchronous commit for performance.

Manual Failover Only:
There is no support for automatic failover. If the primary goes down, you must manually fail over to the secondary.

Phase 1: Network Infrastructure Setup



vpc-02f4ac1f21322daa6 / Manoj_MS_SQL_VPC

Details

VPC ID
vpc-02f4ac1f21322daa6

State
Available

Block Public Access
Off

DNS hostnames
Enabled

DNS resolution
Enabled

Tenancy
default

DHCP option set
dopt-0659267d

Main route table
rtb-00b6de164265ca522

Main network ACL
acl-02610f4e53459ef2d

Default VPC
No

IPv4 CIDR
10.0.0.0/24

IPv6 pool
–

IPv6 CIDR (Network border group)
–

Network Address Usage metrics
Disabled

Route 53 Resolver DNS Firewall rule groups
Failed to load rule groups

Owner ID
031342435657

Resource map

VPC
Your AWS virtual network
Manoj_MS_SQL_VPC

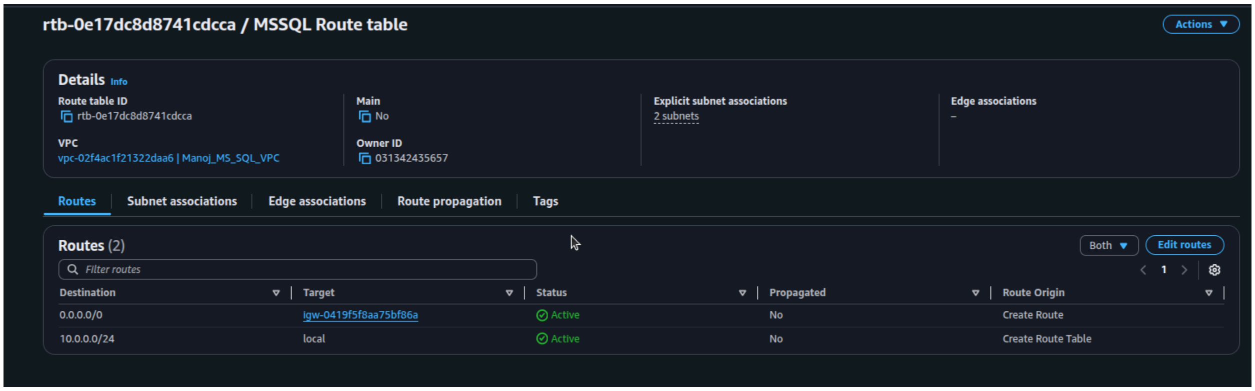
Subnets (2)
Subnets within this VPC
us-east-1a
A Source MSSQL (us east 1a)
us-east-1b
B Destination MSSQL(us east 1b)

Route tables (2)
Route network traffic to resources
MSSQL Route table
rtb-00b6de164265ca522

Network connections (1)
Connections to other networks
MSSQL_IGW

Figure: VPC Screenshot

Summary: The infrastructure is set up in US EAST - 1. With the subnets in us-east-1a and us-east-1b. Both have the same route table redirecting the traffic internally and to the internet as shown in the snip below.



rtb-0e17dc8d8741cdcca / MSSQL Route table

Details Info

Route table ID: rtb-0e17dc8d8741cdcca

Main: No

Explicit subnet associations: 2 subnets

Edge associations: –

VPC: vpc-02f4ac1f21322daa6 | Manoj_MS_SQL_VPC

Owner ID: 031342435657

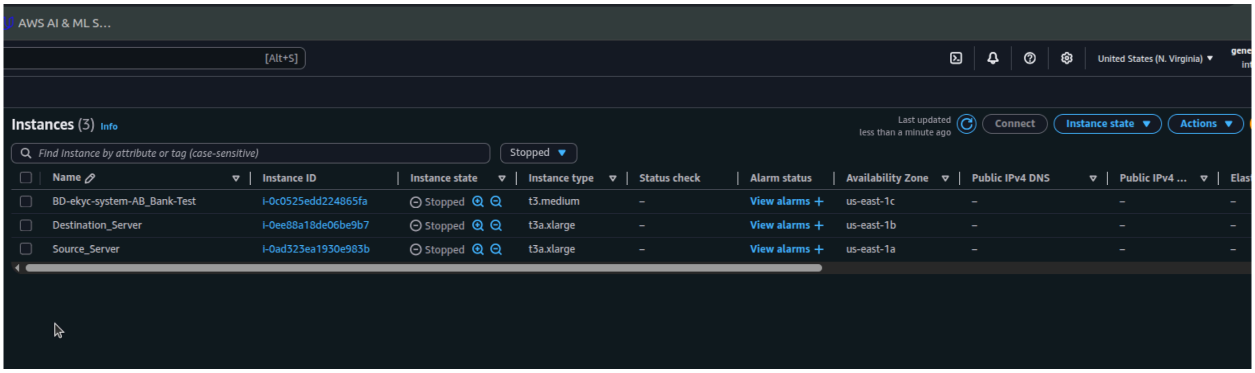
Routes | Subnet associations | Edge associations | Route propagation | Tags
Both ▼
Edit routes
Filter routes

Routes (2)

Destination	Target	Status	Propagated	Route Origin
0.0.0.0/0	igw-0419f5f8aa75bf86a	✓ Active	No	Create Route
10.0.0.0/24	local	✓ Active	No	Create Route Table

Figure: Route table for the VPC

Phase 2: Instance Setup:



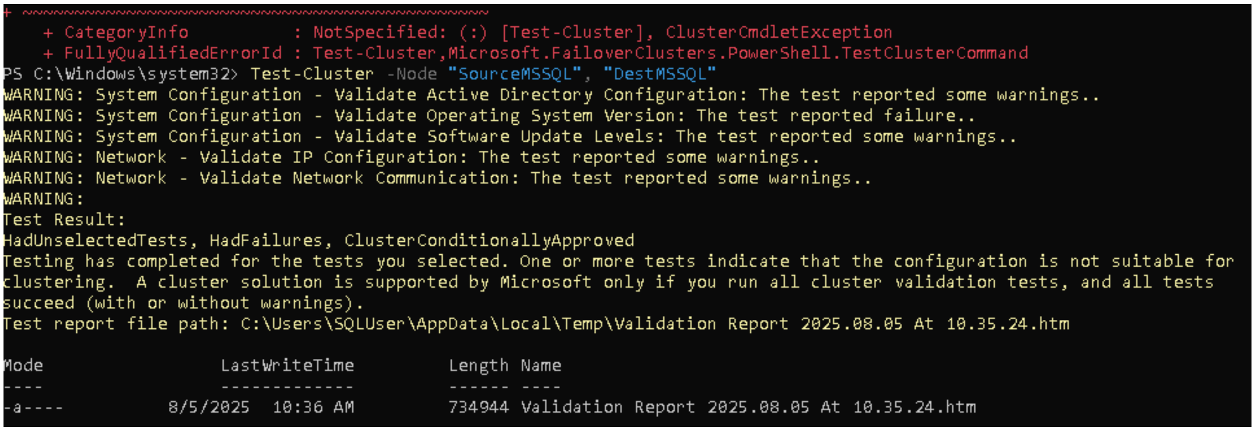
AWS AI & ML S...
[Alt+S]
United States (N. Virginia) ▼
Last updated less than a minute ago
Connect
Instance state ▼
Actions ▼
Find Instance by attribute or tag (case-sensitive)
Stopped ▼

Instances (3) Info

□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elas
□	BD-ekyc-system-AB_Bank-Test	i-0c0525edd224865fa	Stopped	t3.medium	–	View alarms +	us-east-1c	–	–	–
□	Destination_Server	i-0ee88a18de06be9b7	Stopped	t3a.xlarge	–	View alarms +	us-east-1b	–	–	–
□	Source_Server	i-0ad323ea1930e983b	Stopped	t3a.xlarge	–	View alarms +	us-east-1a	–	–	–

Figure: Source and destination MSSQL servers

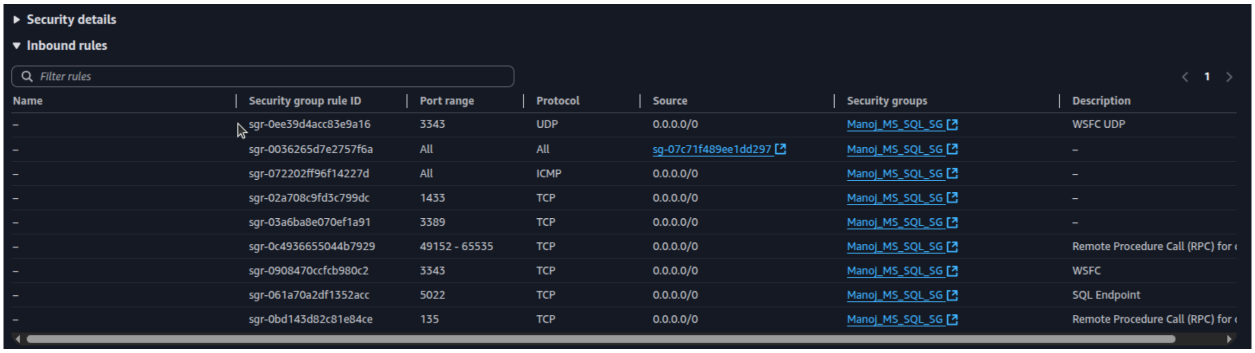
It's a critical note that the two instances that are supposed to be in the Instances must be started **with the same AMI** as we have different checks running while making a cluster, 2 nodes are eligible to be inside a cluster only when they have the same version of operating system. It was a recurring problem during the R&D as shown in the snip below:



```
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : NotSpecified: (:) [Test-Cluster], ClusterCmdletException
    + FullyQualifiedErrorId : Test-Cluster,Microsoft.FailoverClusters.PowerShell.TestClusterCommand
PS C:\Windows\system32> Test-Cluster -Node "SourceMSSQL", "DestMSSQL"
WARNING: System Configuration - Validate Active Directory Configuration: The test reported some warnings..
WARNING: System Configuration - Validate Operating System Version: The test reported failure..
WARNING: System Configuration - Validate Software Update Levels: The test reported some warnings..
WARNING: Network - Validate IP Configuration: The test reported some warnings..
WARNING: Network - Validate Network Communication: The test reported some warnings..
WARNING:
Test Result:
HadUnselectedTests, HadFailures, ClusterConditionallyApproved
Testing has completed for the tests you selected. One or more tests indicate that the configuration is not suitable for
clustering.  A cluster solution is supported by Microsoft only if you run all cluster validation tests, and all tests
succeed (with or without warnings).
Test report file path: C:\Users\SQLUser\AppData\Local\Temp\Validation Report 2025.08.05 At 10.35.24.htm

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----          8/5/2025  10:36 AM         734944 Validation Report 2025.08.05 At 10.35.24.htm
```

The Instances are to be in different subnets for now.

The security group is set up as below:



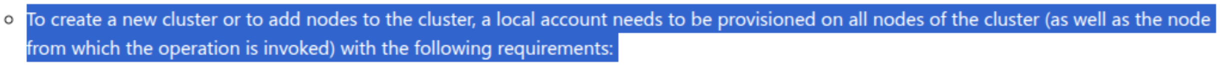
Name	Security group rule ID	Port range	Protocol	Source	Security groups	Description
–	sgr-0ee39d4acc83e9a16	3343	UDP	0.0.0.0/0	Manoj_MS_SQL_SG	WSFC UDP
–	sgr-0036265d7e2757f6a	All	All	sg-07c71f489ee1dd297	Manoj_MS_SQL_SG	–
–	sgr-072202ff96f14227d	All	ICMP	0.0.0.0/0	Manoj_MS_SQL_SG	–
–	sgr-02a708c9fd3c799dc	1433	TCP	0.0.0.0/0	Manoj_MS_SQL_SG	–
–	sgr-03a6ba8e070ef1a91	3389	TCP	0.0.0.0/0	Manoj_MS_SQL_SG	–
–	sgr-0c4936655044b7929	49152 - 65535	TCP	0.0.0.0/0	Manoj_MS_SQL_SG	Remote Procedure Call (RPC) for
–	sgr-0908470ccfcb980c2	3343	TCP	0.0.0.0/0	Manoj_MS_SQL_SG	WSFC
–	sgr-061a70a2df1352acc	5022	TCP	0.0.0.0/0	Manoj_MS_SQL_SG	SQL Endpoint
–	sgr-0bd143d82c81e84ce	135	TCP	0.0.0.0/0	Manoj_MS_SQL_SG	Remote Procedure Call (RPC) for

These rules are nothing less than a mess. It was assumed that the vague ports like 3389, 5022 135 were taking traffic through the public internet but it is not all that necessary. Most likely, we'll only need RDS and rule no 2. **allows all types of traffic (TCP, UDP, ICMP, etc.) on all ports** from any **EC2 instance** that is part of the security group. That is our only requirement since MSSQL server (When used for failover clustering) is known for using weird ports that are not well documented.

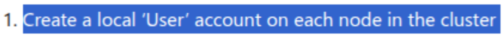
That's all for the EC2 setup, other configurations are carried out normally.

Phase 3: Clustering, Failover Tools, MSSQL Setup

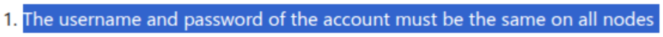
According to this Document, also in the snippets:



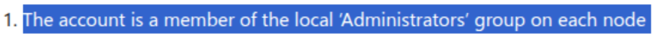
- To create a new cluster or to add nodes to the cluster, a local account needs to be provisioned on all nodes of the cluster (as well as the node from which the operation is invoked) with the following requirements:



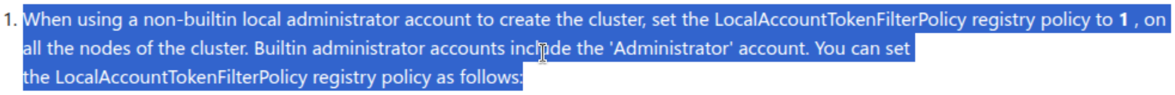
1. Create a local 'User' account on each node in the cluster



1. The username and password of the account must be the same on all nodes

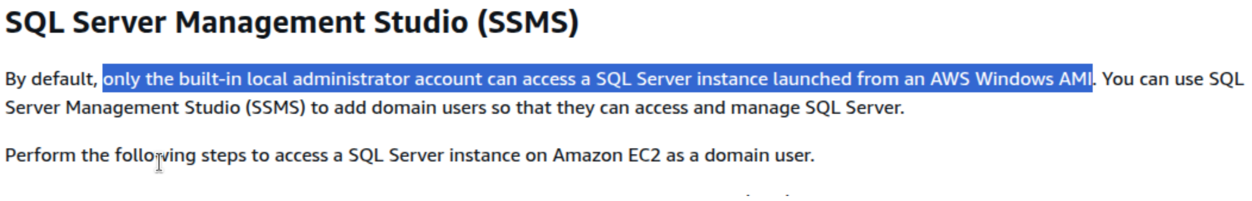


1. The account is a member of the local 'Administrators' group on each node



1. When using a non-builtin local administrator account to create the cluster, set the LocalAccountTokenFilterPolicy registry policy to **1** , on all the nodes of the cluster. Builtin administrator accounts include the 'Administrator' account. You can set the LocalAccountTokenFilterPolicy registry policy as follows:

We have to make new **administrator** accounts with the same passwords in each of the instances that we are treating as nodes. Additionally, they should be assigned to the **Remote Desktop Users** group as sometimes just administrator privilege is not enough for RDP access. Another challenge is:



SQL Server Management Studio (SSMS)

By default, only the built-in local administrator account can access a SQL Server instance launched from an AWS Windows AMI. You can use SQL Server Management Studio (SSMS) to add domain users so that they can access and manage SQL Server.

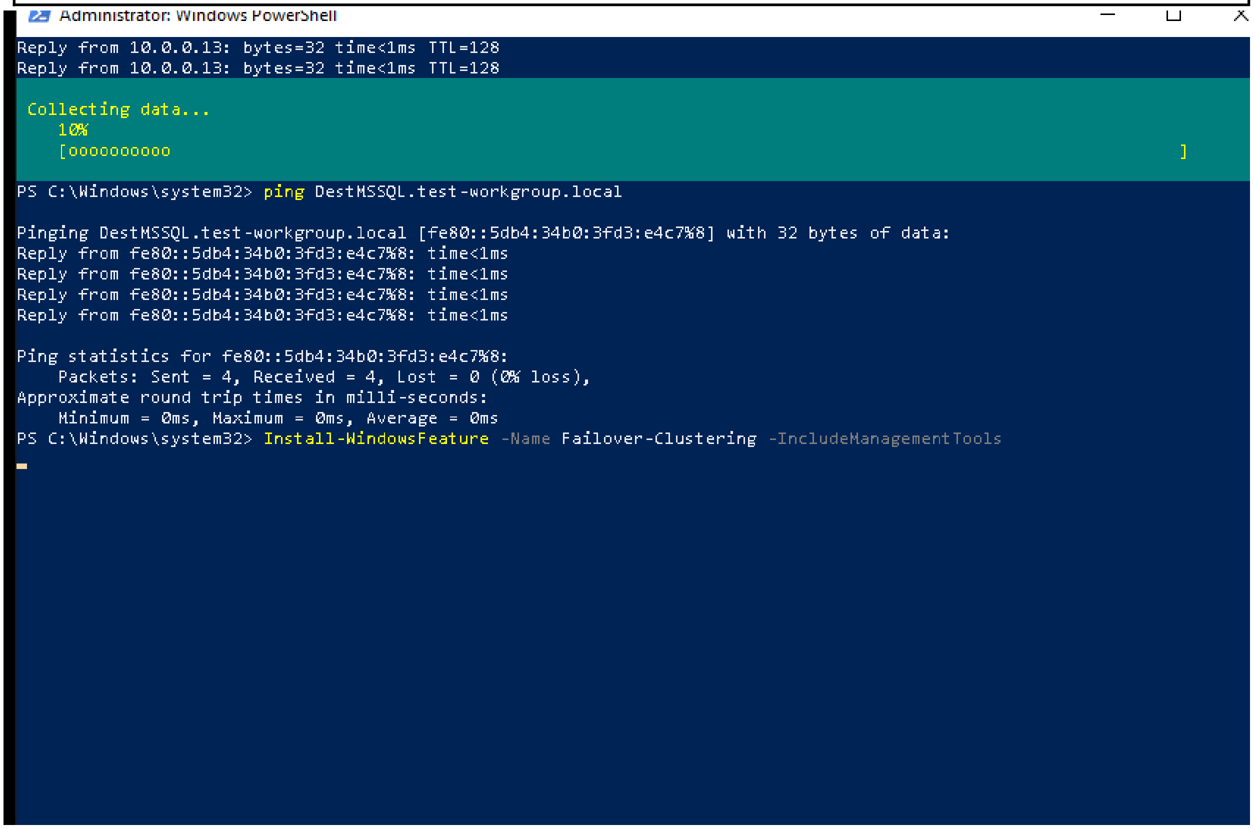
Perform the following steps to access a SQL Server instance on Amazon EC2 as a domain user.

So we should follow this doc for access to new administrator users.

Now for the messy part:

- RDP through the Source Server and Destination server through the **new Administrative user** that we have just created and then install the failover clustering feature using this CLI command. (Note: Also can be done through GUI but CLI faster)

```
Install-WindowsFeature -Name Failover-Clustering -IncludeManagementTools
```



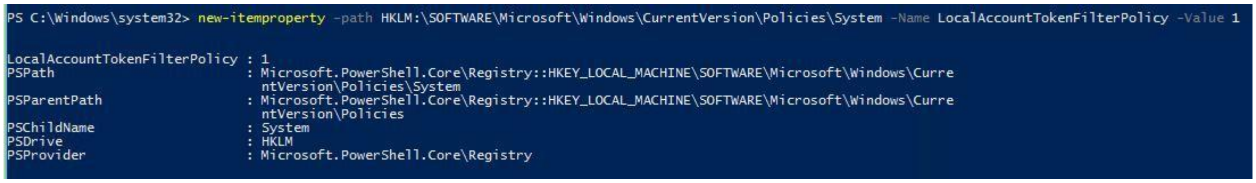
```
Reply from 10.0.0.13: bytes=32 time<1ms TTL=128
Reply from 10.0.0.13: bytes=32 time<1ms TTL=128

 Collecting data...
    10%
    [oooooooooo                                                                                ]

PS C:\Windows\system32> ping DestMSSQL.test-workgroup.local

Pinging DestMSSQL.test-workgroup.local [fe80::5db4:34b0:3fd3:e4c7%8] with 32 bytes of data:
Reply from fe80::5db4:34b0:3fd3:e4c7%8: time<1ms
Reply from fe80::5db4:34b0:3fd3:e4c7%8: time<1ms
Reply from fe80::5db4:34b0:3fd3:e4c7%8: time<1ms
Reply from fe80::5db4:34b0:3fd3:e4c7%8: time<1ms

Ping statistics for fe80::5db4:34b0:3fd3:e4c7%8:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
PS C:\Windows\system32> Install-WindowsFeature -Name Failover-Clustering -IncludeManagementTools
```

It recommends for a restart but hold for a bit

- Also do:

  ```
  new-itemproperty -path
  HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System -Name
  LocalAccountTokenFilterPolicy -Value 1
  ```

  Otherwise it will give error:

  ```
  PS C:\Windows\system32> new-itemproperty -path HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System -Name LocalAccountTokenFilterPolicy -Value 1

  LocalAccountTokenFilterPolicy : 1
  PSPath                        : Microsoft.PowerShell.Core\Registry::HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\Curre
                                  ntVersion\Policies\System
  PSParentPath                  : Microsoft.PowerShell.Core\Registry::HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\Curre
                                  ntVersion\Policies
  PSChildName                   : System
  PSDrive                       : HKLM
  PSProvider                    : Microsoft.PowerShell.Core\Registry
  ```

- Additionally, You have the option to turn off the firewall as whole or run these commands:

  ```
  Enable-NetFirewallRule -DisplayName "File and Printer Sharing (Echo
  Request - ICMPv4-In)"
  Enable-NetFirewallRule -DisplayGroup "Windows Management
  Instrumentation (WMI)"
  ```

- Now this is the right time to restart both instances.

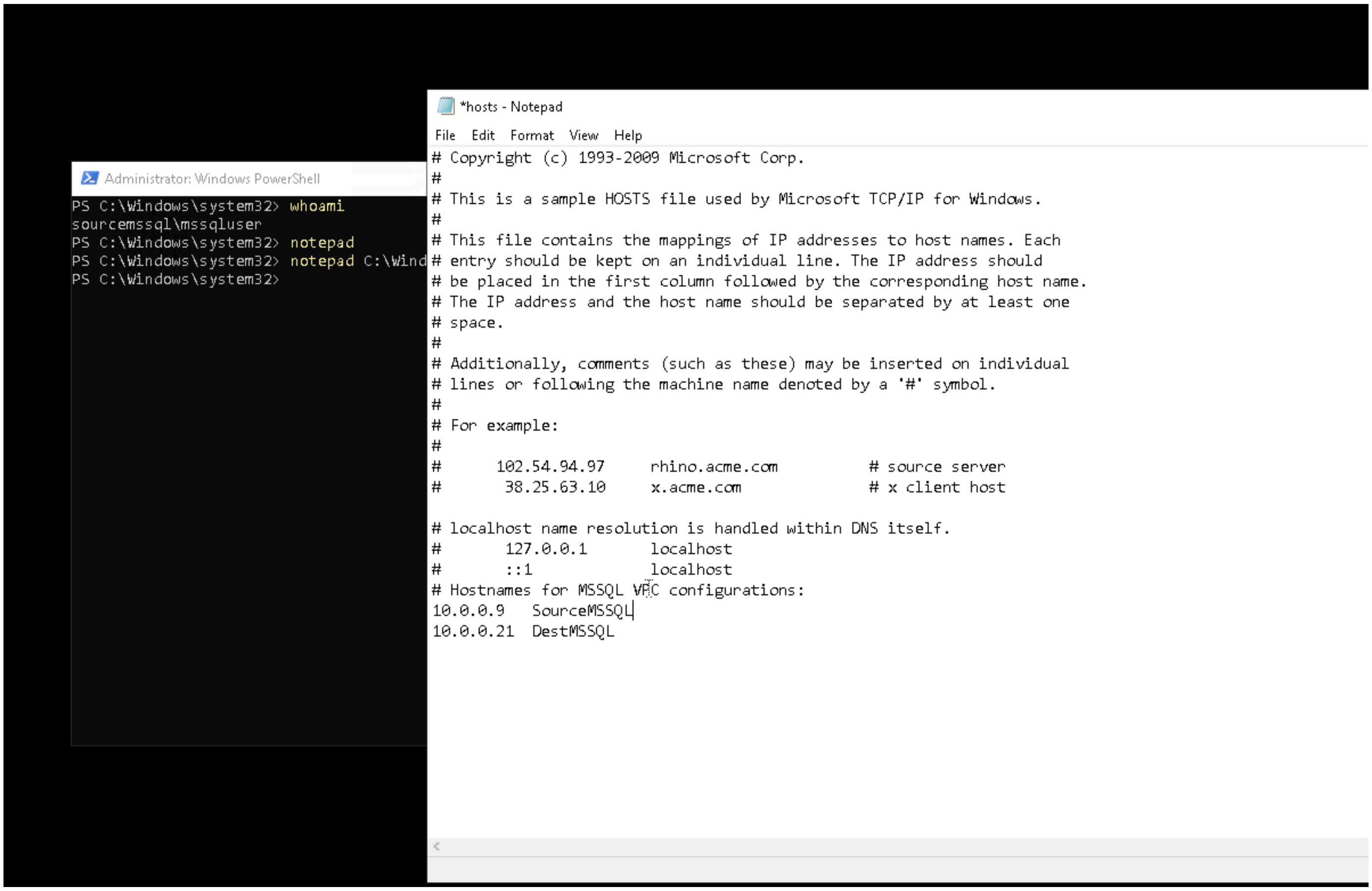
A slight reminder that we are trying to do this setup without setting up an AD, An AD would have been more easier and seamlessly connected to the SQL Server Management Studio (SSMS).

We are trying for resolving hostnames through the hostnames files in the windows path:

```
C:\Windows\System32\Drivers\etc\hosts
```

Simply, in both the instances go to the file and write the private ip address of the two instances, assigned according to the VPC and the subnet's CIDR.

For example:



```
PS C:\Windows\system32> whoami
sourcemssql\mssqluser
PS C:\Windows\system32> notepad
PS C:\Windows\system32> notepad C:\Wind
PS C:\Windows\system32>
```

```
# Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#      102.54.94.97     rhino.acme.com          # source server
#       38.25.63.10     x.acme.com              # x client host

# localhost name resolution is handled within DNS itself.
#       127.0.0.1       localhost
#       ::1             localhost
# Hostnames for MSSQL VPC configurations:
10.0.0.9   SourceMSSQL
10.0.0.21  DestMSSQL
```

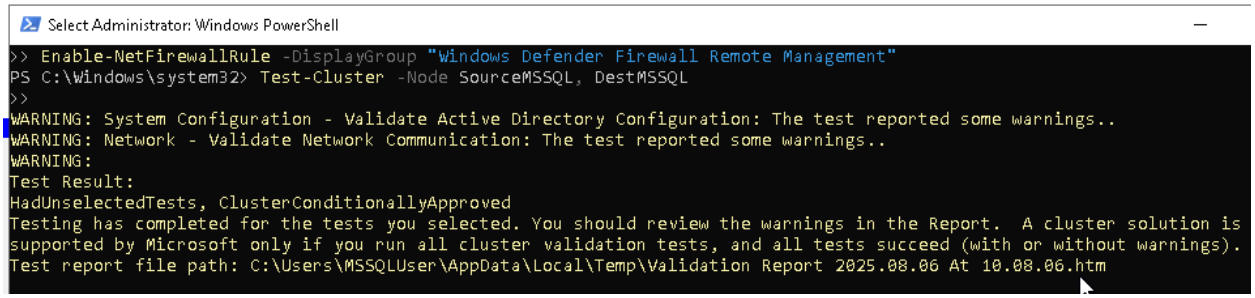
- Try to ping the source server with the destination server and vice versa. This is our crucial step for making the cluster and assigning nodes.
- If everything is completed till this step, we are ready for making failover clusters.

Clusters:

Our main goal for a successful implementation of BAG (Basic Availability Groups) is making clusters and assigning nodes.

- First, we check the compatibility of nodes to make a cluster. We do this by using the following command:

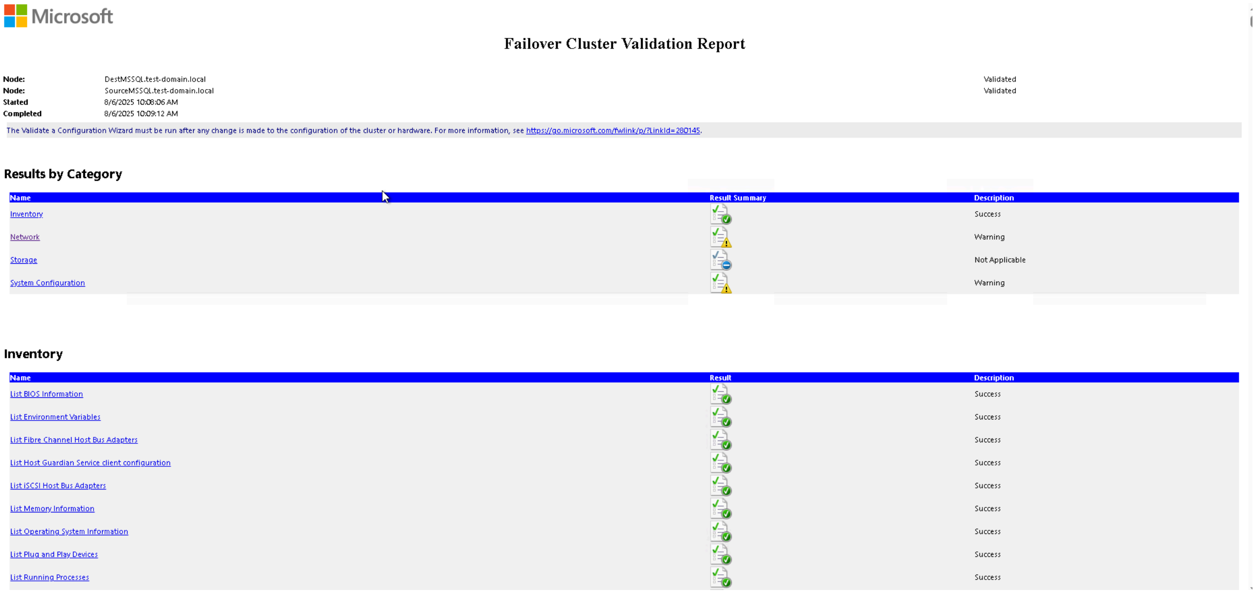
  ```
  Test-Cluster -Node hostname1, hostname2
  ```



Select Administrator: Windows PowerShell

```
>> Enable-NetFirewallRule -DisplayGroup "Windows Defender Firewall Remote Management"
PS C:\Windows\system32> Test-Cluster -Node SourceMSSQL, DestMSSQL
>>
WARNING: System Configuration - Validate Active Directory Configuration: The test reported some warnings..
WARNING: Network - Validate Network Communication: The test reported some warnings..
WARNING:
Test Result:
HadUnselectedTests, ClusterConditionallyApproved
Testing has completed for the tests you selected. You should review the warnings in the Report.  A cluster solution is
supported by Microsoft only if you run all cluster validation tests, and all tests succeed (with or without warnings).
Test report file path: C:\Users\MSSQLUser\AppData\Local\Temp\Validation Report 2025.08.06 At 10.08.06.htm
```

Above command runs several specific checks and makes a detailed report:



Microsoft

Failover Cluster Validation Report

Node:	DestMSSQL.test-domain.local	Validated
Node:	SourceMSSQL.test-domain.local	Validated
Started	8/6/2025 10:08:06 AM	
Completed	8/6/2025 10:09:12 AM	

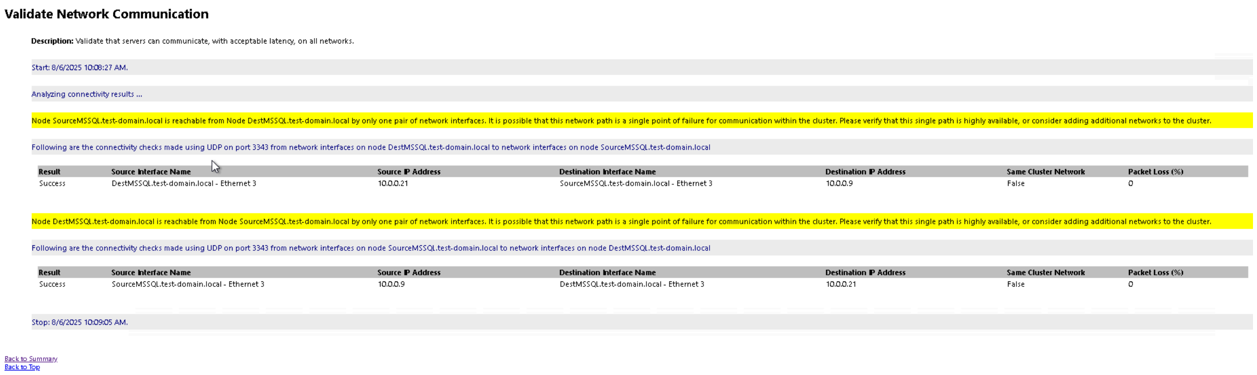
The Validate a Configuration Wizard must be run after any change is made to the configuration of the cluster or hardware. For more information, see https://go.microsoft.com/fwlink/p/?LinkId=280145.

Results by Category

Name	Result Summary	Description
Inventory	✓	Success
Network	⚠	Warning
Storage	⊖	Not Applicable
System Configuration	⚠	Warning

Inventory

Name	Result	Description
List BIOS Information	✓	Success
List Environment Variables	✓	Success
List Fibre Channel Host Bus Adapters	✓	Success
List Host Guardian Service client configuration	✓	Success
List iSCSI Host Bus Adapters	✓	Success
List Memory Information	✓	Success
List Operating System Information	✓	Success
List Plug and Play Devices	✓	Success
List Running Processes	✓	Success



Validate Network Communication

Description: Validate that servers can communicate, with acceptable latency, on all networks.

Start: 8/6/2025 10:08:27 AM.

Analyzing connectivity results ...

Node SourceMSSQL.test-domain.local is reachable from Node DestMSSQL.test-domain.local by only one pair of network interfaces. It is possible that this network path is a single point of failure for communication within the cluster. Please verify that this single path is highly available, or consider adding additional networks to the cluster.

Following are the connectivity checks made using UDP on port 3343 from network interfaces on node DestMSSQL.test-domain.local to network interfaces on node SourceMSSQL.test-domain.local

Result	Source Interface Name	Source IP Address	Destination Interface Name	Destination IP Address	Same Cluster Network	Packet Loss (%)
Success	DestMSSQL.test-domain.local - Ethernet 3	10.0.0.21	SourceMSSQL.test-domain.local - Ethernet 3	10.0.0.9	False	0

Node DestMSSQL.test-domain.local is reachable from Node SourceMSSQL.test-domain.local by only one pair of network interfaces. It is possible that this network path is a single point of failure for communication within the cluster. Please verify that this single path is highly available, or consider adding additional networks to the cluster.

Following are the connectivity checks made using UDP on port 3343 from network interfaces on node SourceMSSQL.test-domain.local to network interfaces on node DestMSSQL.test-domain.local

Result	Source Interface Name	Source IP Address	Destination Interface Name	Destination IP Address	Same Cluster Network	Packet Loss (%)
Success	SourceMSSQL.test-domain.local - Ethernet 3	10.0.0.9	DestMSSQL.test-domain.local - Ethernet 3	10.0.0.21	False	0

Stop: 8/6/2025 10:09:05 AM.

Back to Summary
Back to Top

It is recommended to check the warnings thoroughly and then only continue, otherwise it will result in a **broken cluster**.

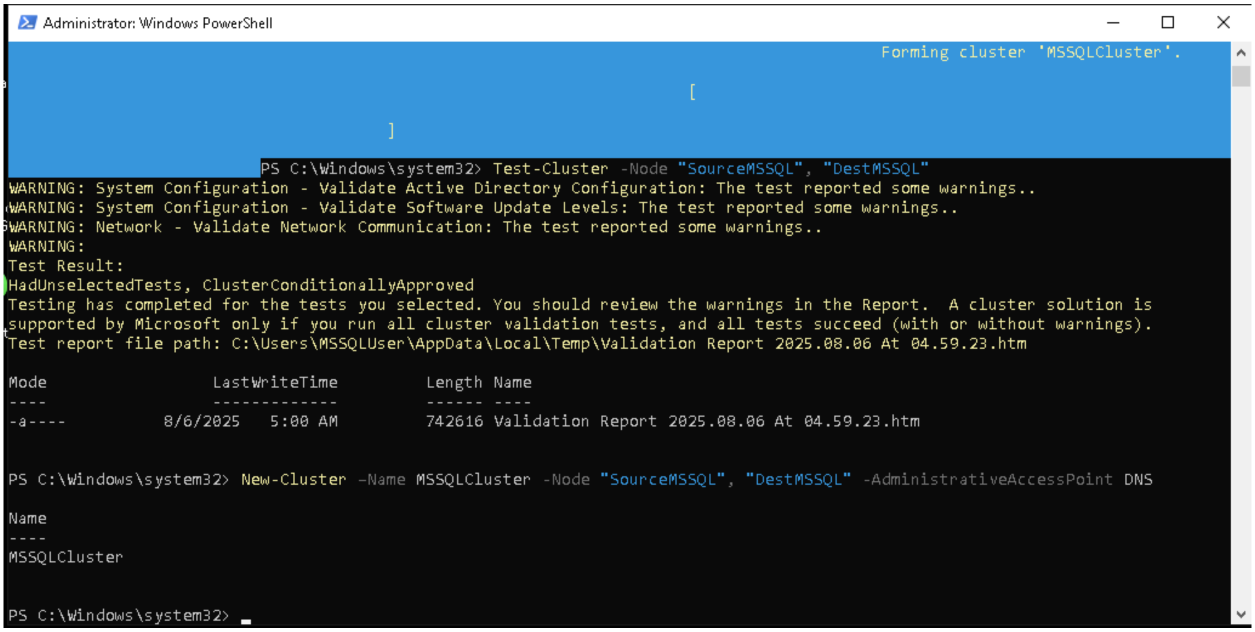
- We now continue with the configuration and make a cluster using powershell. Remember to:
  1. Run the powershell as **administrator**
  2. Make sure that you are doing RDS with the new local administrator account, which we created.
  3. All above steps are completely cleared without issues.

- For making cluster, we use the command (syntax):

```
New-Cluster -Name MSSQLCluster -Node SourceMSSQL,DestMSSQL
-AdministrativeAccessPoint Dns -StaticAddress 10.0.1.100 -NoStorage
```

Syntax Breakdown

Parameter	**Description**
New-Cluster	The command that creates the failover cluster.
-Name	Specifies the network name for the cluster.
-Node	A comma-separated list of servers to add to the cluster.
-AdministrativeAccessPoint	Dns creates the cluster for a workgroup (no Active Directory).
-StaticAddress	Assigns a static IP address to the cluster's network name.
-NoStorage	Creates the cluster without shared storage, ideal for SQL Always On groups.



```
                                                                                         Forming cluster 'MSSQLCluster'.
                                                                                       [
                                              ]
                          PS C:\Windows\system32> Test-Cluster -Node "SourceMSSQL", "DestMSSQL"
WARNING: System Configuration - Validate Active Directory Configuration: The test reported some warnings..
WARNING: System Configuration - Validate Software Update Levels: The test reported some warnings..
WARNING: Network - Validate Network Communication: The test reported some warnings..
WARNING:
Test Result:
HadUnselectedTests, ClusterConditionallyApproved
Testing has completed for the tests you selected. You should review the warnings in the Report.  A cluster solution is
supported by Microsoft only if you run all cluster validation tests, and all tests succeed (with or without warnings).
Test report file path: C:\Users\MSSQLUser\AppData\Local\Temp\Validation Report 2025.08.06 At 04.59.23.htm

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----          8/6/2025   5:00 AM         742616 Validation Report 2025.08.06 At 04.59.23.htm


PS C:\Windows\system32> New-Cluster -Name MSSQLCluster -Node "SourceMSSQL", "DestMSSQL" -AdministrativeAccessPoint DNS

Name
----
MSSQLCluster


PS C:\Windows\system32> _
```

Figure: Running the New-Cluster command

Also to check if the nodes are assigned in the cluster we can run the command:

```
Get-Cluster
```

These are to be run in both nodes.

For more details on resources like ips and storage we use:

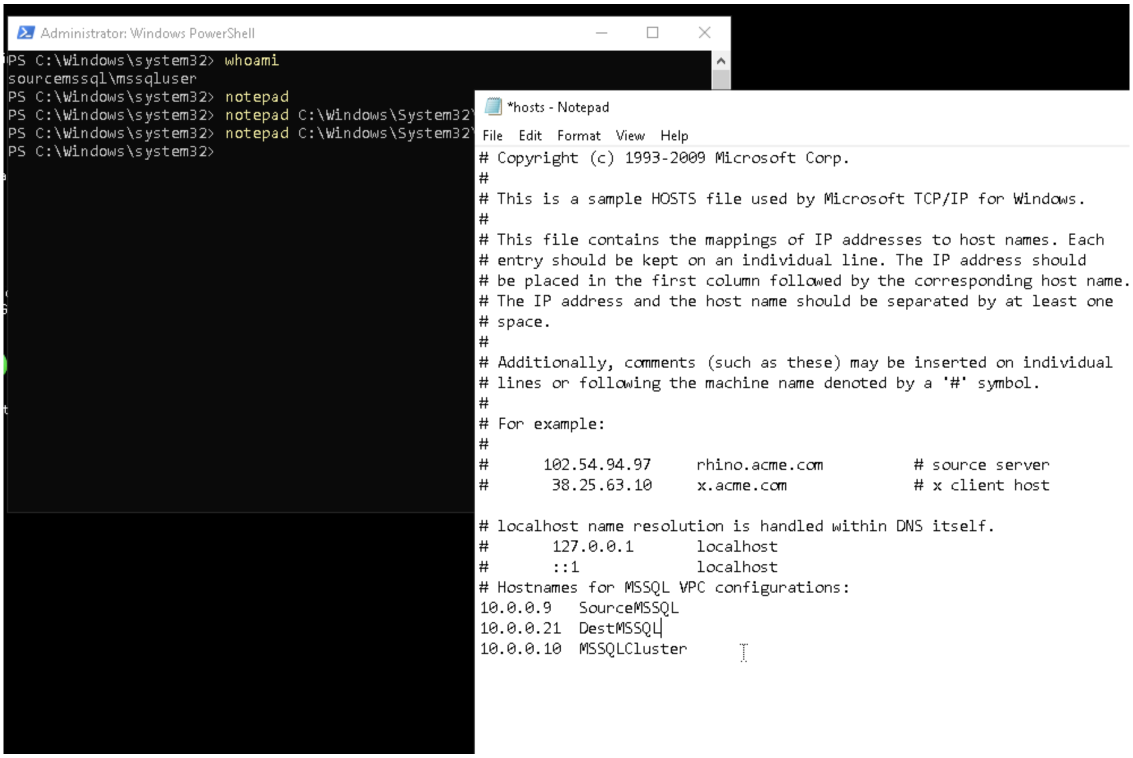
```
Get-ClusterResource
```

If both IPs are offline in the cluster, remote registry service fails sometimes, run these individually:

```
Get-Service -Name RemoteRegistry
Start-Service -Name RemoteRegistry
Set-Service -Name RemoteRegistry -StartupType Automatic
```

Now we add the cluster name that we just made to the hostname file:

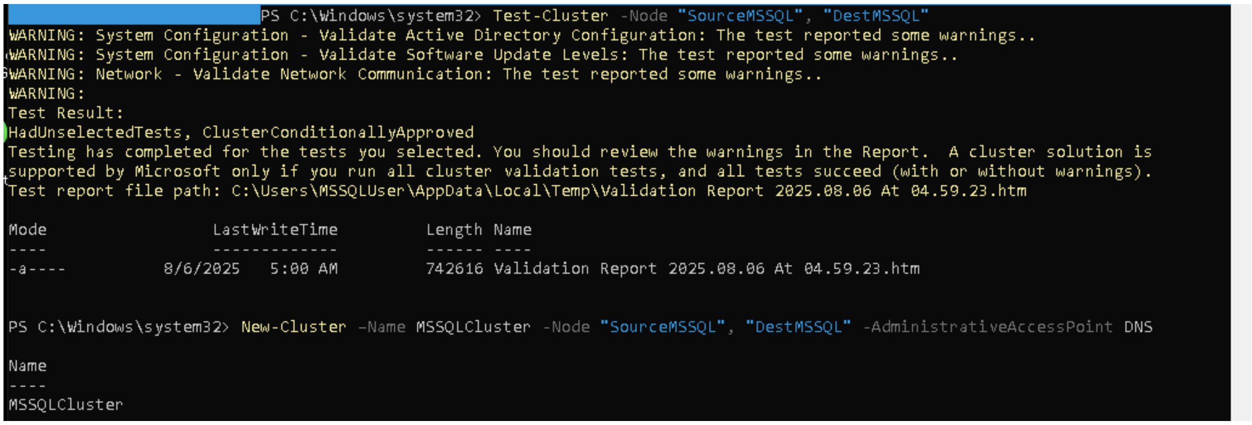
```
C:\Windows\System32\Drivers\etc\hosts
```



```
PS C:\Windows\system32> whoami
sourcemssql\mssqluser
PS C:\Windows\system32> notepad
PS C:\Windows\system32> notepad C:\Windows\System32`
PS C:\Windows\system32> notepad C:\Windows\System32`
PS C:\Windows\system32>
```

```
# Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#      102.54.94.97     rhino.acme.com          # source server
#       38.25.63.10     x.acme.com              # x client host

# localhost name resolution is handled within DNS itself.
#       127.0.0.1       localhost
#       ::1             localhost
# Hostnames for MSSQL VPC configurations:
10.0.0.9   SourceMSSQL
10.0.0.21  DestMSSQL
10.0.0.10  MSSQLCluster
```

Then we ping the cluster. In my case the command is, `ping MSSQLCluster`



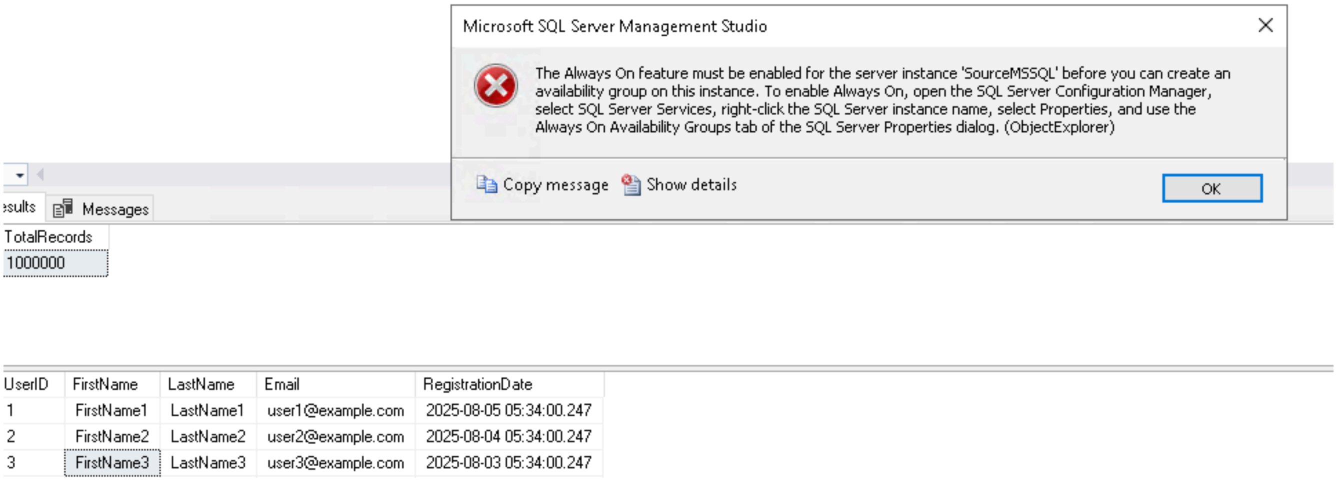
```
PS C:\Windows\system32> Test-Cluster -Node "SourceMSSQL", "DestMSSQL"
WARNING: System Configuration - Validate Active Directory Configuration: The test reported some warnings..
WARNING: System Configuration - Validate Software Update Levels: The test reported some warnings..
WARNING: Network - Validate Network Communication: The test reported some warnings..
WARNING:
Test Result:
HadUnselectedTests, ClusterConditionallyApproved
Testing has completed for the tests you selected. You should review the warnings in the Report.  A cluster solution is
supported by Microsoft only if you run all cluster validation tests, and all tests succeed (with or without warnings).
Test report file path: C:\Users\MSSQLUser\AppData\Local\Temp\Validation Report 2025.08.06 At 04.59.23.htm

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----          8/6/2025    5:00 AM         742616 Validation Report 2025.08.06 At 04.59.23.htm


PS C:\Windows\system32> New-Cluster -Name MSSQLCluster -Node "SourceMSSQL", "DestMSSQL" -AdministrativeAccessPoint DNS

Name
----
MSSQLCluster
```

Important Note: *In my above screenshot that is running the New-Cluster line, i do not have any static IP, but the IP can be seen using the* `Get-ClusterResource` *command in the powershell.*

Cluster is successfully configured.

Phase 4: Making Availability Groups

Have a look at this SSMS screenshots:



The Always On feature must be enabled for the server instance 'SourceMSSQL' before you can create an availability group on this instance. To enable Always On, open the SQL Server Configuration Manager, select SQL Server Services, right-click the SQL Server instance name, select Properties, and use the Always On Availability Groups tab of the SQL Server Properties dialog. (ObjectExplorer)

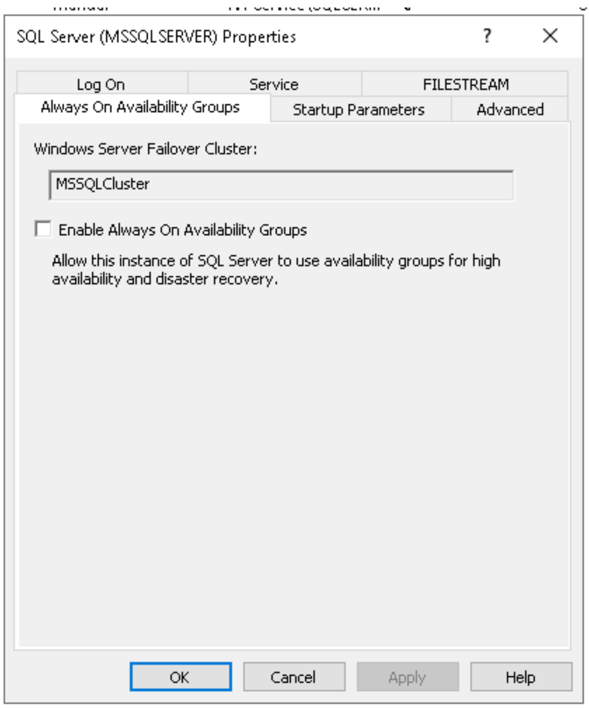
TotalRecords
1000000

UserID	FirstName	LastName	Email	RegistrationDate
1	FirstName1	LastName1	user1@example.com	2025-08-05 05:34:00.247
2	FirstName2	LastName2	user2@example.com	2025-08-04 05:34:00.247
3	FirstName3	LastName3	user3@example.com	2025-08-03 05:34:00.247

So we go and enable availability groups
We go to the path : *Server Configuration Manager --> SQL Server Services --> SQL Server name and Right click to properties*

Then we just tick the option: Enable Always on Availability Groups

If the node is not in the cluster, It will give the error like below:
"The computer is not in a failover cluster"
If the cluster is not broken, this will not happen

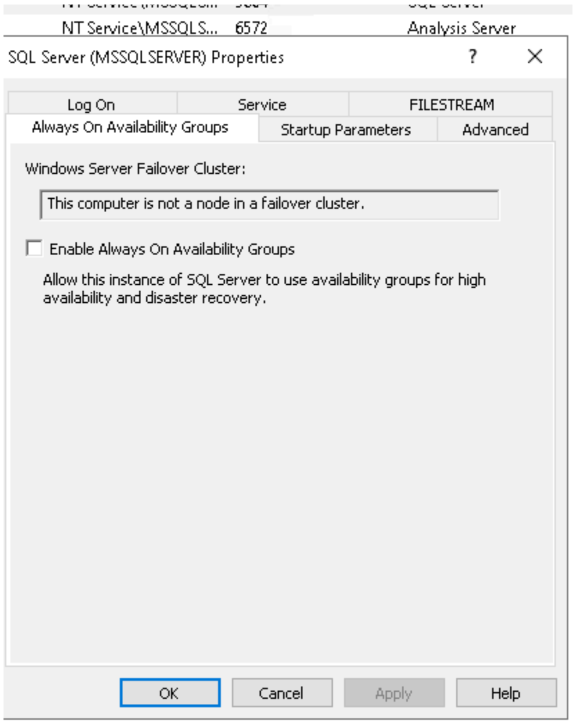


SQL Server (MSSQLSERVER) Properties

Windows Server Failover Cluster: MSSQLCluster

☐ Enable Always On Availability Groups

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



NT Service\MSSQLS... 6572 Analysis Server

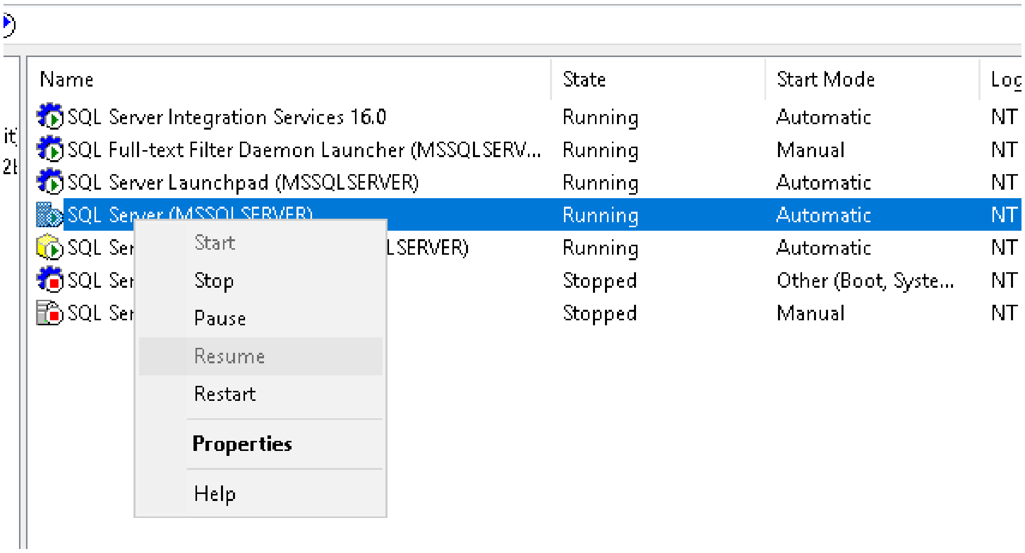
SQL Server (MSSQLSERVER) Properties

Windows Server Failover Cluster: This computer is not a node in a failover cluster.

☐ Enable Always On Availability Groups

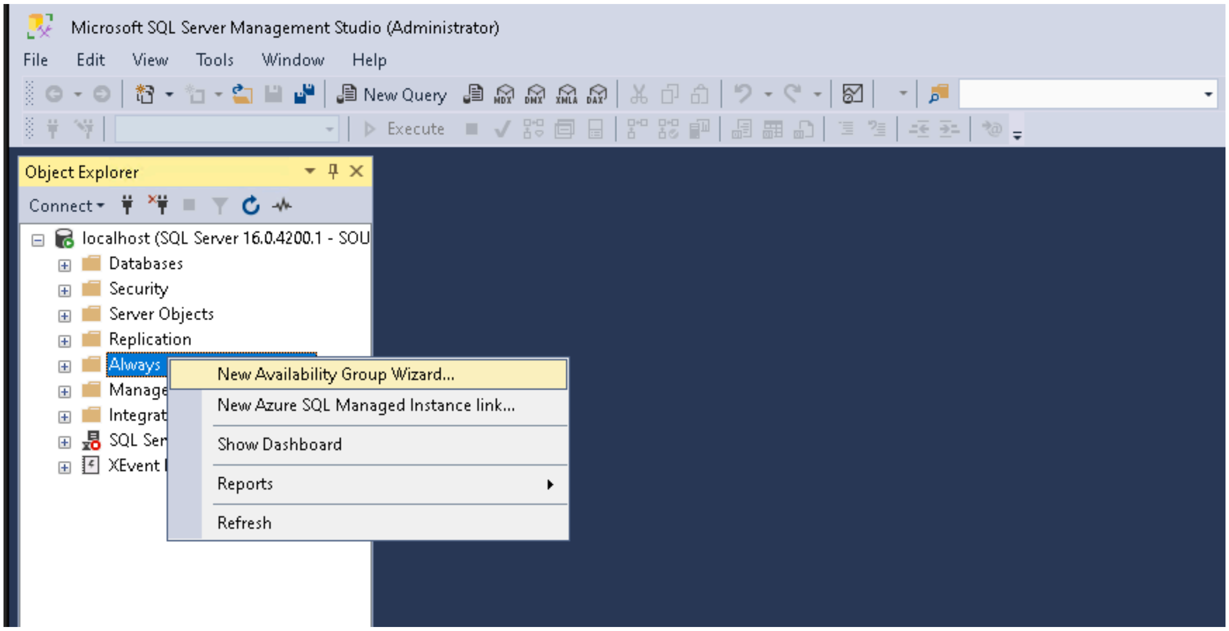
Allow this instance of SQL Server to use availability groups for high availability and disaster recovery.

Then Restart the MSSQL server.

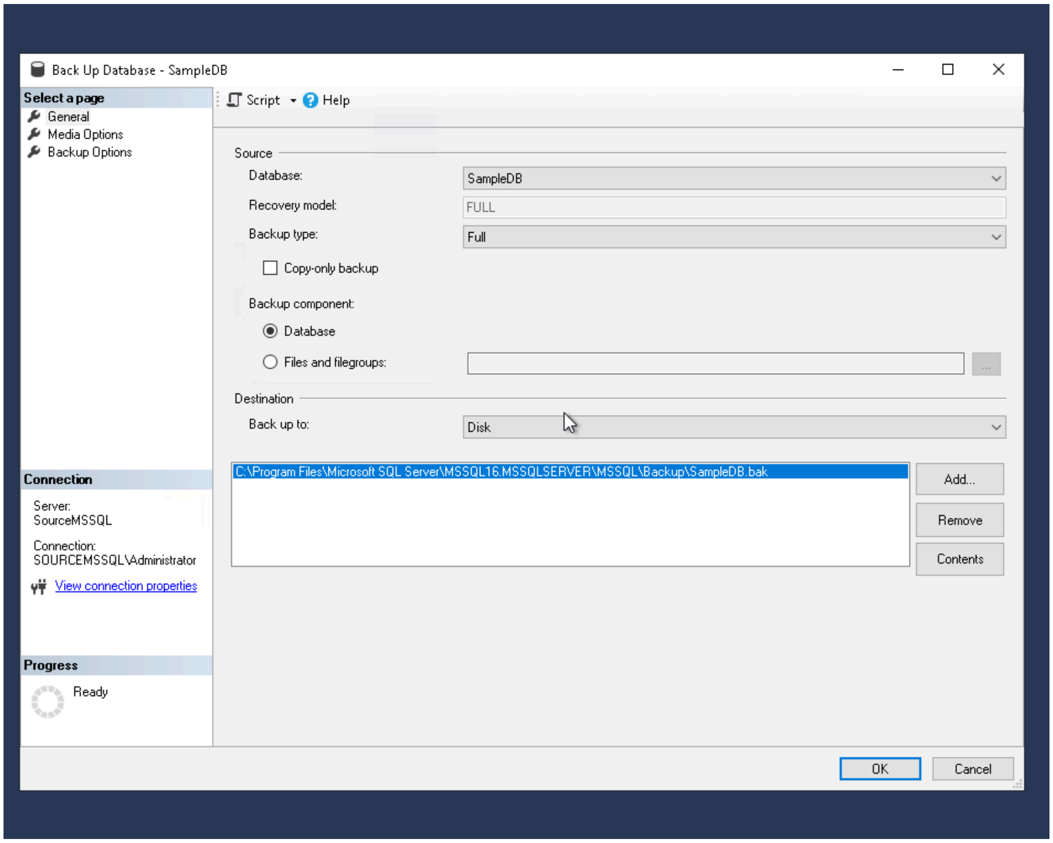


Name	State	Start Mode	Log
SQL Server Integration Services 16.0	Running	Automatic	NT
SQL Full-text Filter Daemon Launcher (MSSQLSERV...	Running	Manual	NT
SQL Server Launchpad (MSSQLSERVER)	Running	Automatic	NT
SQL Server (MSSQLSERVER)	Running	Automatic	NT
SQL Ser LSERVER)	Running	Automatic	NT
SQL Ser	Stopped	Other (Boot, Syste...	NT
SQL Ser	Stopped	Manual	NT

Now head over to the SQL Studio Management. Make a new Availability group using the GUI.


Microsoft SQL Server Management Studio (Administrator)
File Edit View Tools Window Help
New Query
Execute
Object Explorer
Connect
localhost (SQL Server 16.0.4200.1 - SOU
Databases
Security
Server Objects
Replication
Always
Manage
Integrat
SQL Ser
XEvent
New Availability Group Wizard...
New Azure SQL Managed Instance link...
Show Dashboard
Reports
Refresh

Critical note here: By default the BAG supports only one database in the availability group. For the database to be used like that, It MUST be fully backed up. The process is shown below.



Source

Database: SampleDB

Recovery model: FULL

Backup type: Full

☐ Copy-only backup

Backup component:

● Database

○ Files and filegroups:

Destination

Back up to: Disk

C:\Program Files\Microsoft SQL Server\MSSQL16.MSSQLSERVER\MSSQL\Backup\SampleDB.bak

Add... Remove Contents

Connection

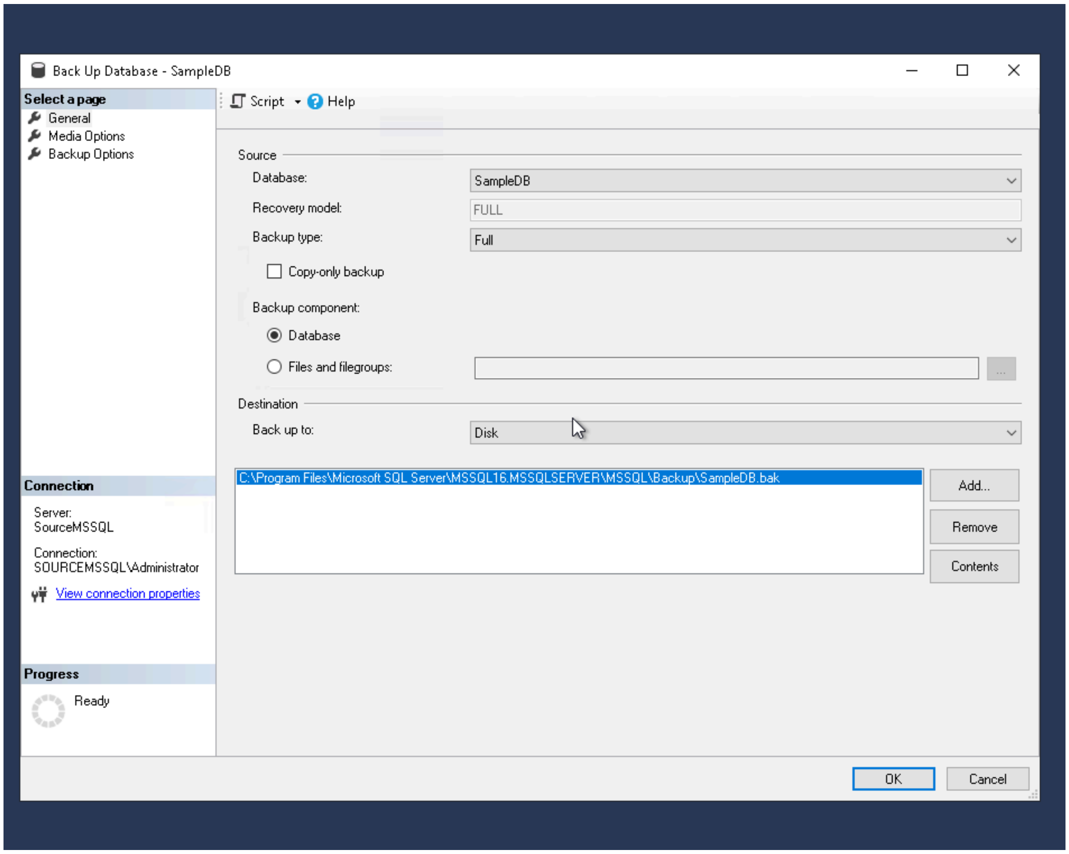
Server:
SourceMSSQL

Connection:
SOURCEMSSQL\Administrator

View connection properties

Progress

Ready



Database: SampleDB

Recovery model: FULL

Backup type: Full

□ Copy-only backup

Backup component:

● Database

○ Files and filegroups:

Back up to: Disk

C:\Program Files\Microsoft SQL Server\MSSQL16.MSSQLSERVER\MSSQL\Backup\SampleDB.bak

Connection

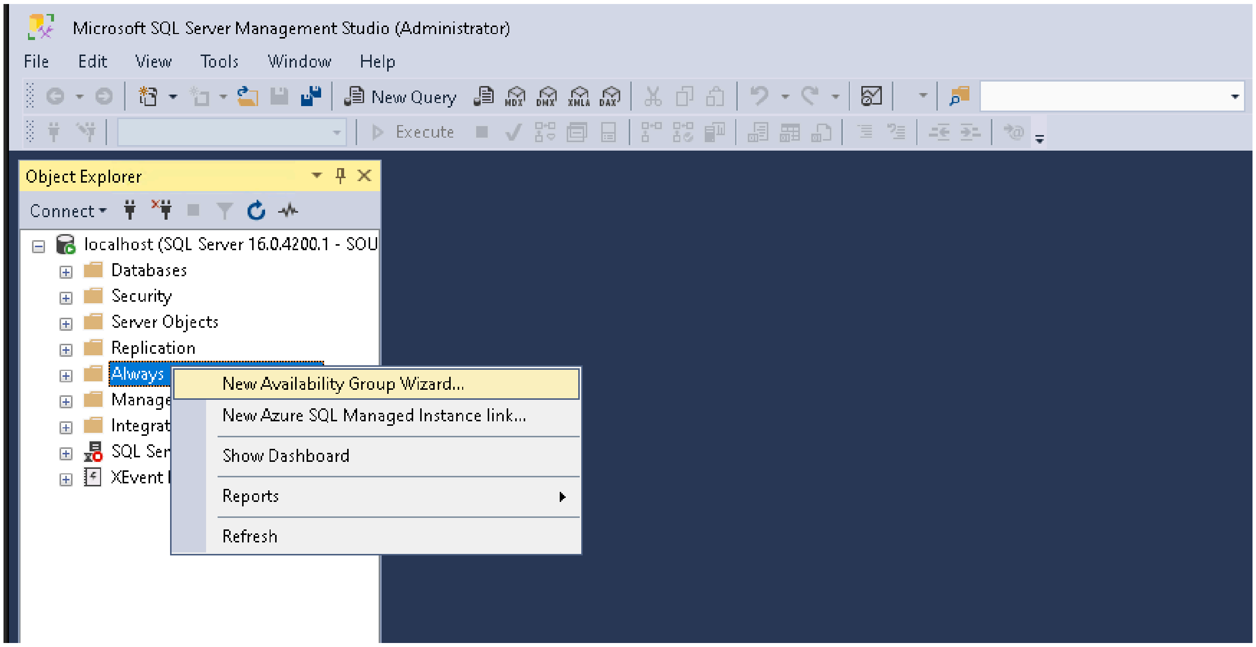
Server:
SourceMSSQL

Connection:
SOURCEMSSQL\Administrator

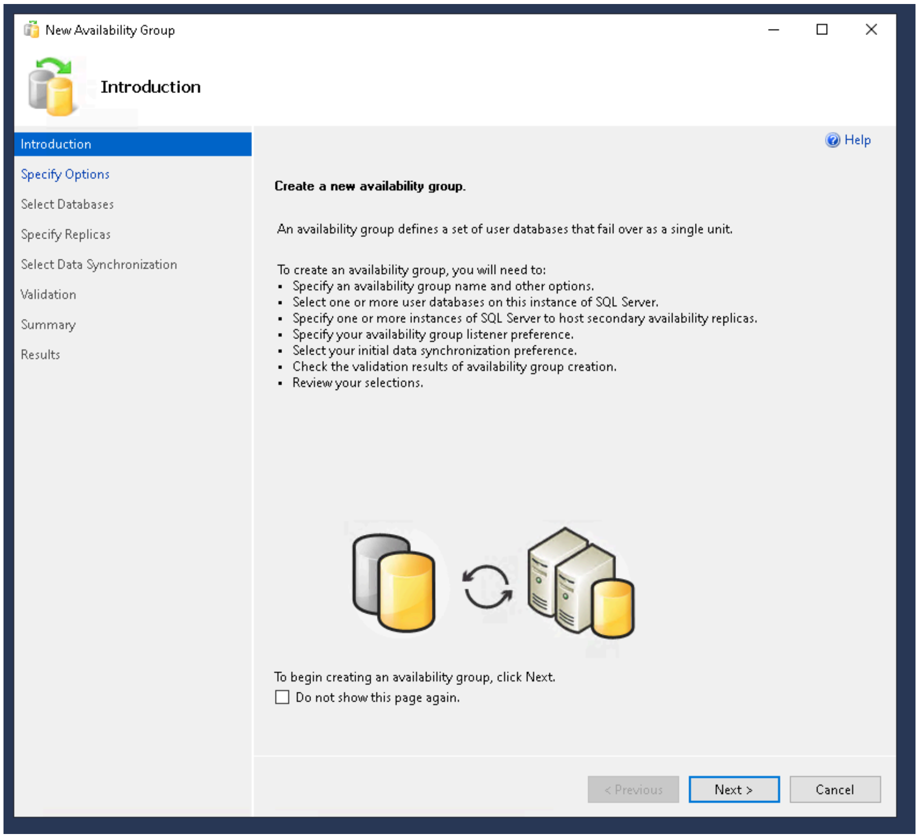
Progress

Ready

If the database is backed up, We'll continue here:



Select the new availability group wizard



Create a new availability group.

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

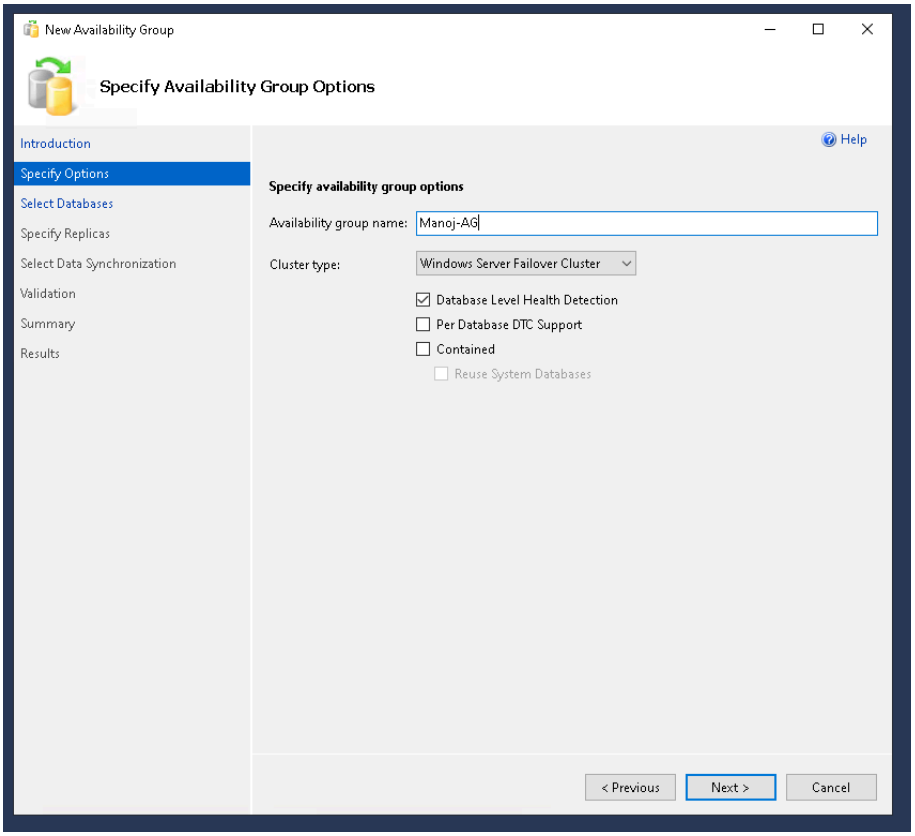
To create an availability group, you will need to:

- 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.

Click Next



Specify Availability Group Options

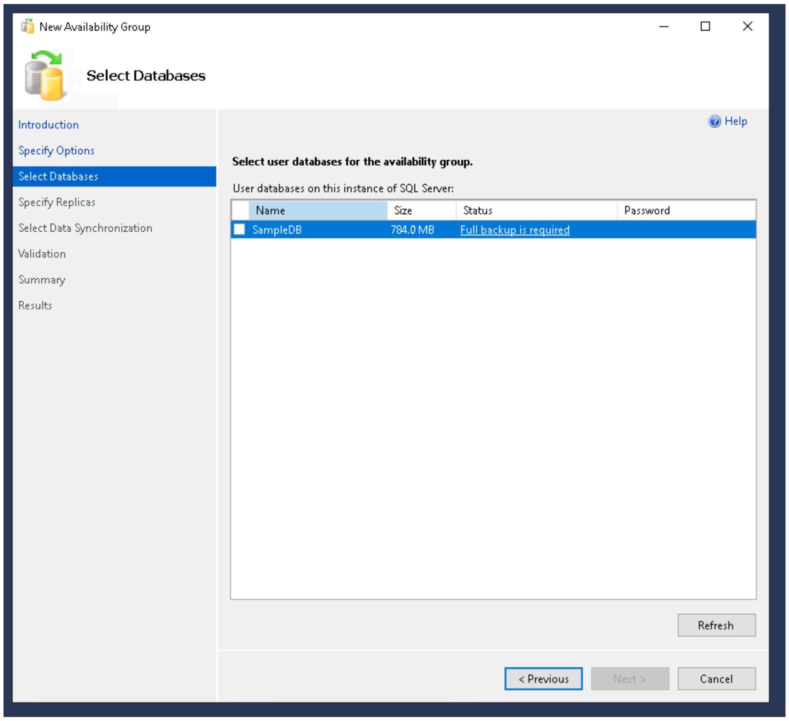
Specify availability group options

Availability group name: Manoj-AG

Cluster type: Windows Server Failover Cluster

☑ Database Level Health Detection
☐ Per Database DTC Support
☐ Contained
☐ Reuse System Databases

Set Availability Group’s name



Select Databases

Select user databases for the availability group.

User databases on this instance of SQL Server:

	Name	Size	Status	Password
☐	SampleDB	784.0 MB	Full backup is required	

Select the Database

This completes the BAG setup for our scenario !

Refer to the next doc (AWS Hybrid AD setup) for the next part. (⊙人⊙)

Additional References:

https://www.youtube.com/watch?v=CbXtHGBVBjU

https://techcommunity.microsoft.com/blog/failoverclustering/workgroup-and-multi-domain-clusters-in-windows-server-2016/372059

https://docs.aws.amazon.com/sql-server-ec2/latest/userguide/connect-sql-server-on-ec2-instance.html