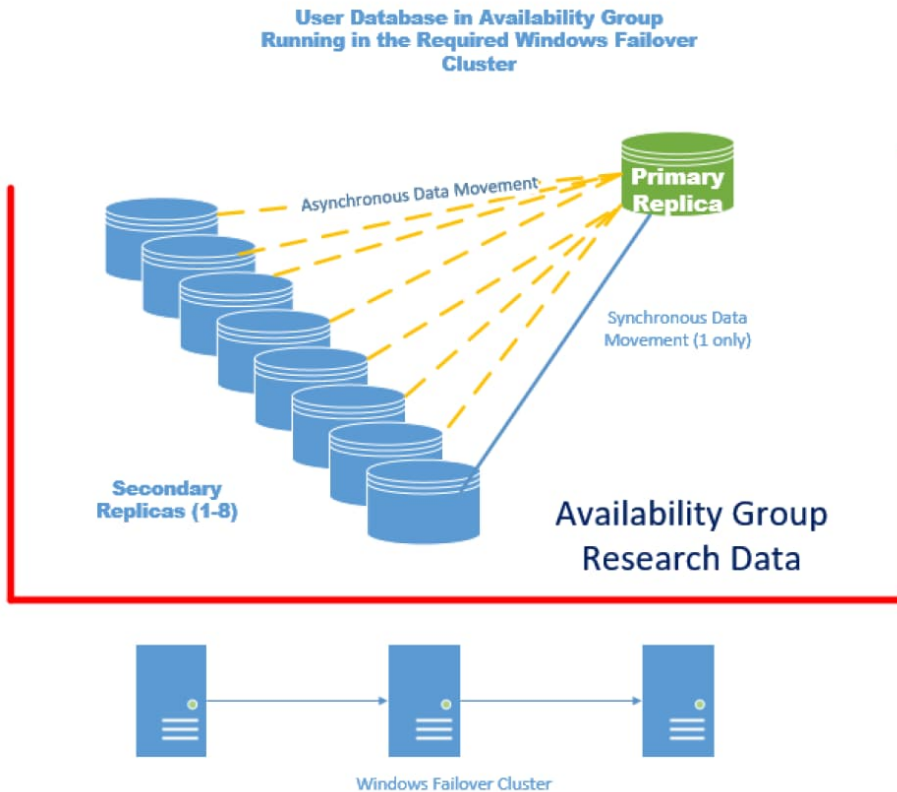


# MODULE 5 SQL 2016 ALWAYS ON AVAILABILITY GROUPS:

## LESSONS

- Availability Groups Concepts
- Availability Groups and Replicas
- Primary Replica
- Secondary Replica
- Availability Group Listener
- Availability Mode
- Failover Modes
- Automatic Failover Requirements
- Topologies
- SSMS and Availability Groups
- The Dashboard
- Cluster Manager for Availability Groups
- Flexible Failover Policy
- Replicating Logins and Jobs
- Partially Contained Databases

## AVAILABILITY GROUPS AND REPLICAS



An availability group solution supports a set of primary user databases and one to eight sets of corresponding secondary user databases. The secondary databases are kept up to date with the primary databases by transferring transaction log blocks from each primary database to every secondary database either synchronously or asynchronously over the network.

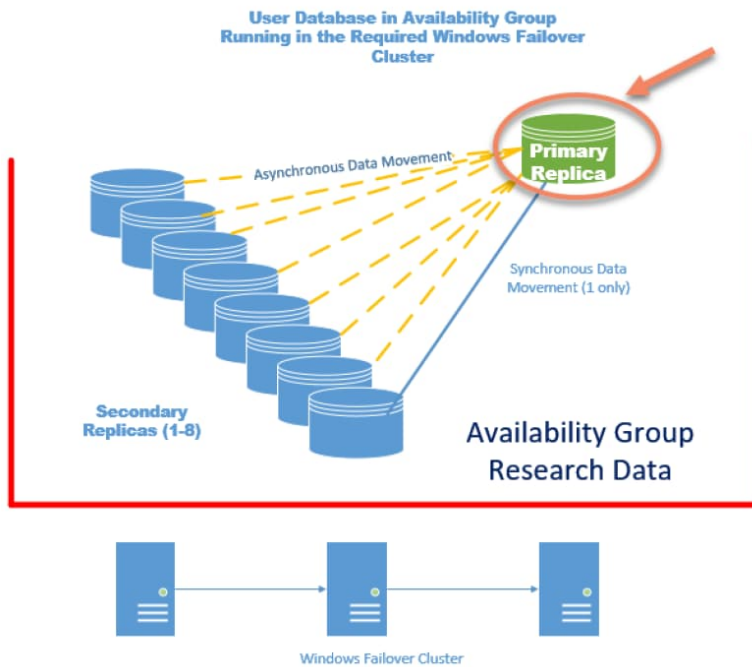
The concept of the Availability Group is a container similar to the concept of containers in Windows Linux and Docker. The contained user databases and failover as a unit. Only the user databases are in the container not external resources like logins, jobs, and linked servers. Those assets do not fail over with the user databases so they need to already be on the secondary replicas.

The failover partners running on the other instances of SQL are called availability replicas. They host a copy of the availability user databases.

It is required that these availability replicas be hosted on a separate instance of SQL running on a different node of Windows Server Failover Cluster. Each SSQL instance can host multiple availability groups but each availability group can only have a single copy of a user database. This sounds obvious but with containers it is sometimes misunderstood that they make a hard separation in a SQL instance just like SQL instances make a hard separation on the SQL server but they do not.

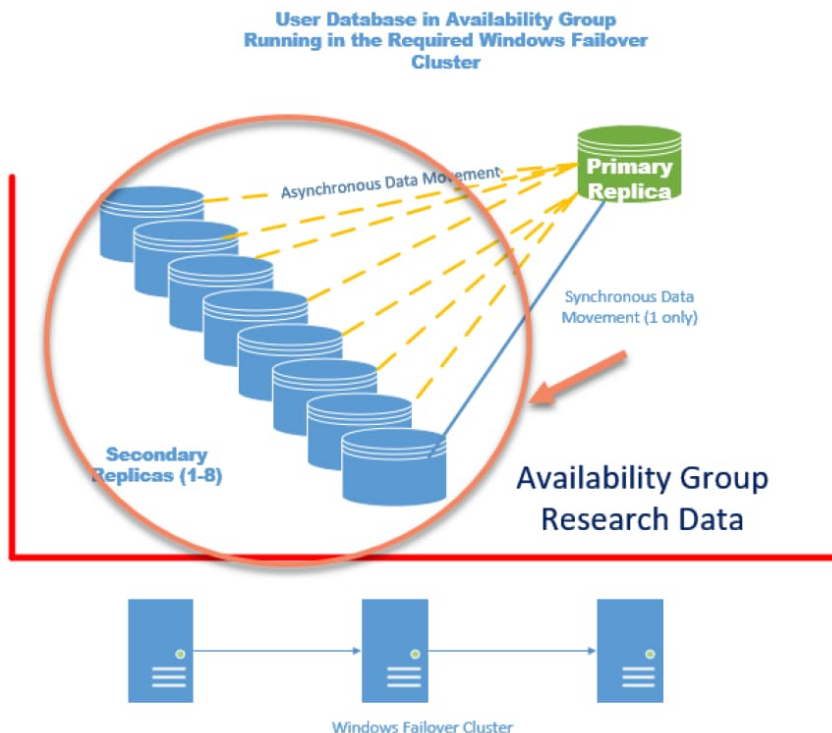
Be careful to distinguish the requirement of Windows cluster which is required and SSQL server Failover Cluster which is optional.

## PRIMARY REPLICA



In Availability groups one container has its property set to Primary replica. Think of the Availability group as a collection of replicas. Each replica has settings in its properties and the container itself which is the availability group has settings that control the settings of the individual replicas. So in an availability group (container) one replica is allowed to set its property to primary and only one just like only one secondary replica can set its property for data synchronization to synchronous.

## SECONDARY REPLICAS



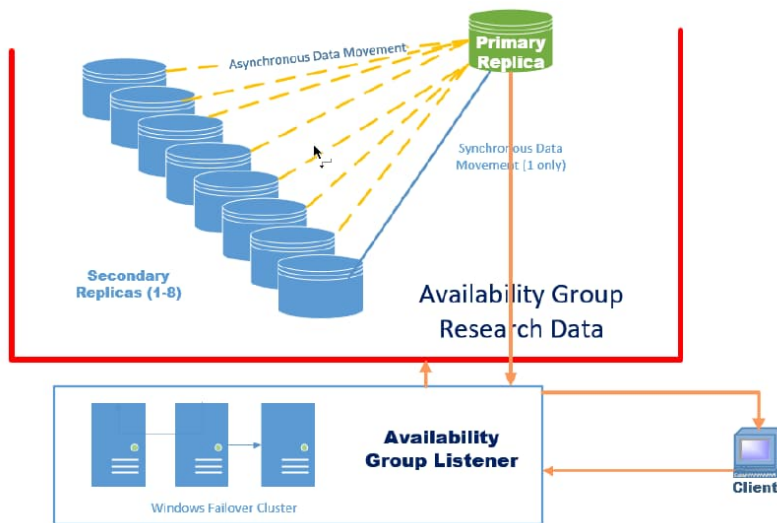
A Secondary Replica maintains a copy of each primary database in the availability group. These replicas are potential failover targets. A secondary replica can be an active replica for read only access.

Availability groups are nothing like the concept of Merge replication where each server can allow read/write.

If a secondary replica is readable it has three possible settings.

- Allows all connections with read-only.
- Disallows all connections which is the default.
- Allows connections with the property 'ApplicationIntent- ReadOnly'

## AVAILABILITY GROUP LISTENER



An availability group Listener is an abstraction that allows the clients to connect to a resource hosted in the Windows Server Cluster similar to how the clients connected to the SQL FCI. This facilitates the seamless transfer of the connection to a secondary replica if we have primary replica failure.

A Listener is a cluster resource with:

- A VNN
- VIP address or multiple
- Availability group dependency
- DNS Name
- Port number

## AVAILABILITY MODE

The availability mode of each availability replica determines whether the primary replica waits to commit the transactions on the database until the secondary replica has written the transactions log record to disk.

## Synchronous Commit Mode

In synchronous-commit mode, the primary replica waits to send the transaction confirmation to the clients until the secondary replica writes the transaction log records to disk. This mode allows zero data loss at the cost of increased transaction latency. A limit of three secondary replicas can operate in this mode. Synchronous commit supports both automatic and manual failover (next topic).

## Asynchronous Commit Mode

In asynchronous-commit mode, the primary replica does not wait for the secondary replica to write the transaction log records to disk. The primary replica sends the transaction confirmation to the clients as soon as the transaction log blocks are persisted on the primary database. All of the replicas can operate in this mode. Asynchronous mode supports only manual failover.

## FAILOVER MODES

An availability group failover is a process during which the failover target takes over the primary role, recovers the databases, and brings them online as the new primary databases. When the former primary is available, it takes over the secondary role and its databases become the secondary database.

## AUTOMATIC FAILOVER WITHOUT DATA LOSS

occurs on the loss of the primary replica. Automatic failover causes a synchronized secondary replica to take over the primary replica role with guaranteed data protection.

## AUTOMATIC FAILOVER REQUIREMENTS:

- Both primary and secondary replicas are running synchronous commit.
- Mode is set to automatic
- WSF has a quorum
- The flexible failover policy for the availability group (not yet covered) meets a failover condition
- The secondary replica is synchronized with the primary.

## MANUAL

Planned Manual without data loss occurs when an administrator manually initiates the failover to cause a synchronized secondary replica to take over the primary replica role with guaranteed data protection.

Forced Manual failover with potential data loss is the only type of failover that an administrator can initiate when the primary replica is lost and the secondary replica is not failover ready or no secondary replica is synchronized. This is the only form of failover supported by asynchronous-commit replicas because they are never synchronized.

Forced manual failover is a disaster recovery option that allows an administrator to use a secondary replica as a warm standby replica after a failure occurs.

## MANUAL FAILOVER REQUIREMENTS

- Both the primary and secondary replicas have the synchronous commit mode
- The secondary is synchronized with the primary
- WSFC has a quorum

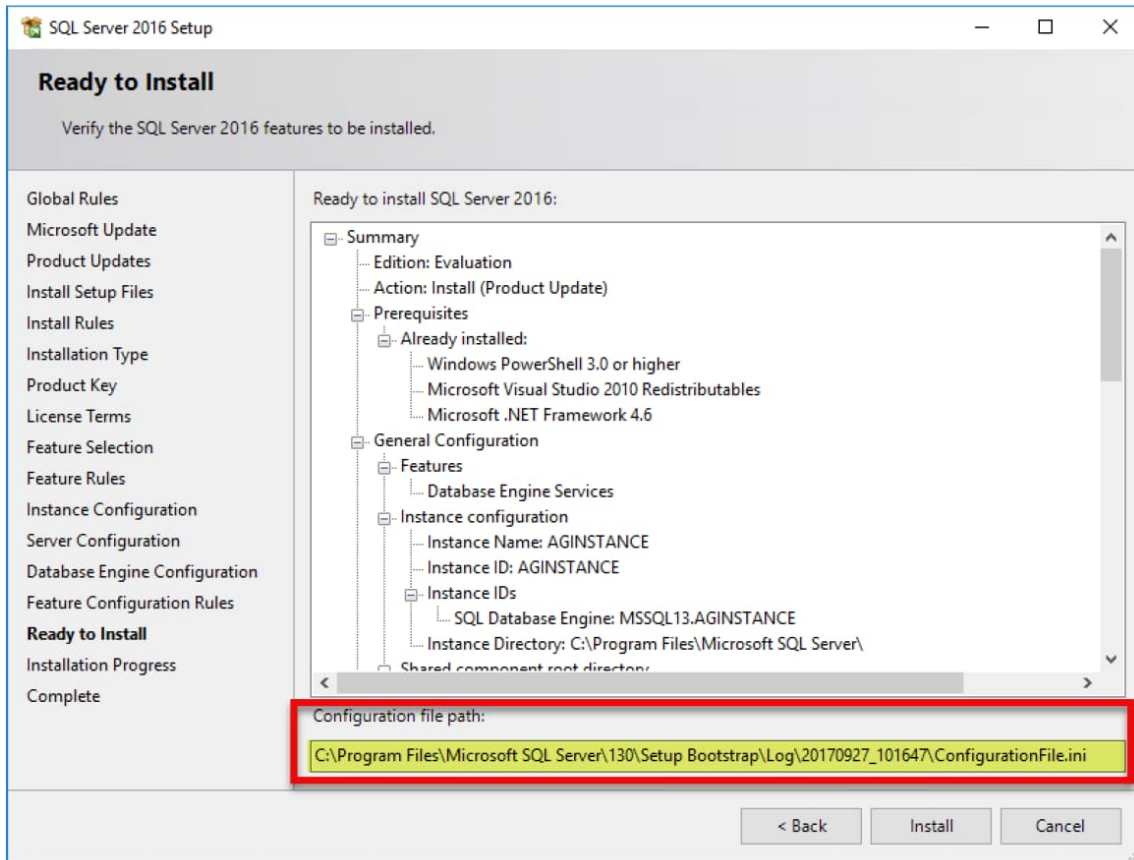
## COMMON TOPOLOGIES

1. **Stand Alone to Stand Alone**  
which is the most common and simplest. As the SQL servers are standalone there is no shared storage. As this is usually implemented in the same data center it can be enhanced by adding a third node and the availability secondary replica to a separate data center.
2. **SQL FCI to Stand Alone**  
This brings part of the advantage of SQL FCI to availability groups.
3. **SQL FCI to SQL FCI**  
This adds the advantages of SQL FCI to both groups.

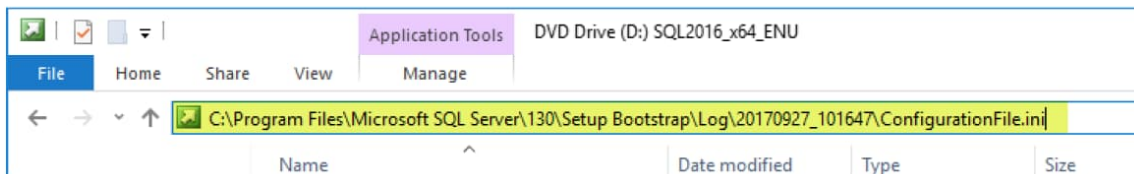
## LAB: CREATE A SQL INSTANCE FOR THE AVAILABILITY GROUP

1. Using **Hyper-V Manager**, switch to **SQL1** virtual machine (if logon is required, log on as Student).
2. Use **File Explorer** to start the **SQL setup.exe**.
3. In the **SQL Server Installation Center** dialog box, click the **Installation** tab.
4. Click **New SQL Server stand-alone installation or add features to an existing installation**.
5. In the **Microsoft Update** dialog box, click **Next**.
6. In the **Installation Type** dialog box, leave **Perform a new installation of SQL Server 2016** selected, and click **Next**.
7. In **Product Key** dialog box, click **Next**.
8. In the **License Terms** dialog box, accept the license terms, then click **Next**.
9. In the **Feature Selection** dialog box, place a check in the **Database Engine Services** check box.
10. Click **Next**.
11. In **Instance Configuration** dialog box, move to the **Named instance** text box and enter AGInstance.

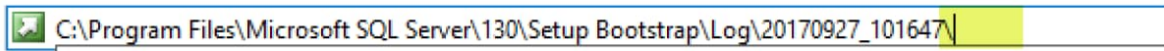
12. Click **Next**.
13. In the **Server Configuration** dialog box, click **Next**.
14. In the **Database Engine Configuration** dialog box, click the radio button to use **Mixed Mode**.
15. Enter and confirm Password in the two **password** text boxes.
16. Click **Add Current User**.
17. Click **Next**.
18. In the **Ready to Install** dialog box, before clicking **Install**, copy the path to the configuration file.



19. Switch to **File Explorer**.
20. **Paste** the address you copied into the address bar in **File Explorer**.



21. Remove the file name so that the address ends with \.

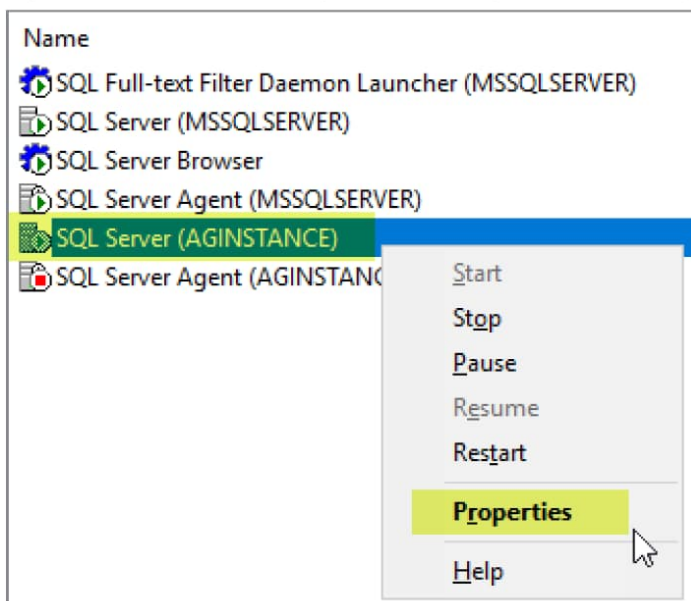


22. Press **Enter** and review the results noticing you were taken to the location of the **ConfigurationFile.ini** file.
23. Switch back to **SQL Server 2016 Setup** and click **Install**.
24. In the **Complete** dialog box, review the results, then click **Close**.
25. Close the **SQL Server Installation Center** dialog box.
26. Switch to **File Explorer**, then locate and **Copy** the **ConfigurationFile.ini** file.
27. Using **Hyper-V Manager**, switch to **SQL2** and **SQL3** virtual machines and **Paste** the **ConfigurationFile.ini** file on the **Desktop** of each.
28. Starting with **SQL2**, use **Hyper-V Manager** to switch to **SQL2**.
29. Use **File Explorer** to start the **SQL setup.exe**.
30. In the **SQL Server Installation Center** dialog box, click the **Advanced** tab.
31. Click the **Install based on configuration file** link.
32. In the **Select Template File** dialog box, navigate to the **Desktop** and double-click to select the **ConfigurationFile.ini** file you pasted there.
33. In the **Microsoft Update** dialog box, click **Next**.
34. In the **Product Updates** dialog box, click **Next**.
35. In the **Installation Type** dialog box, click **Next**.
36. In **Product Key** dialog box, click **Next**.
37. In the **License Terms** dialog box, accept the license terms, then click **Next**.
38. In the **Feature Selection** dialog box, click **Next**.
39. In **Instance Configuration** dialog box, click **Next**.
40. In the **Server Configuration** dialog box, click **Next**.
41. In the **Database Engine Configuration** dialog box, enter and confirm Passw0rd in the two **password** text boxes.
42. Click **Next**.
43. In the **Ready to Install** dialog box, click **Install**.
44. In the **Complete** dialog box, review the results, then click **Close**.
45. Close the **SQL Server Installation Center** dialog box.

46. Use **Hyper-V Manager** to switch to **SQL3** and repeat.

## LAB: ENABLE THE SQL SERVER ALWAYS ON AVAILABILITY GROUP FEATURE

1. Using **Hyper-V Manager**, switch to **SQL1** virtual machine (if logon is required, log on as Student).
2. Press the **Windows** key, then scroll through and locate **Microsoft SQL Server 2016**.
3. Expand **Microsoft SQL Server 2016**.
4. Locate and click **SQL Server 2016 Configuration Manager**.
5. In the **User Account Control** dialog box, click **Yes**.
6. Move to the pane on the left and click to select **SQL Server Services** tab.
7. Right-click on **SQL Server (AGINSTANCE)**, then click **Properties**.



8. In the **SQL Server (AGINSTANCE) Properties** dialog box, click the **AlwaysOn High Availability** tab.
9. Place a check in the **Enable AlwaysOn Availability Groups** check box.
10. Click **Apply**.
11. In the **Warning** dialog box advising **Any changes made will be saved**, click **OK**.
12. Click the **Log On** tab.
13. Navigate to the **Log on as** section, and notice **This account** radio button is selected.
14. Click **Browse**.

15. In the **Select User or Group** dialog box, move to the **Enter the object name to select** text box and enter AOHA\Student.
16. Click **Check Names**.
17. Click **OK**.
18. Move to the password text boxes, then enter and confirm Passw0rd.
19. Click **Apply**.
20. In the **Confirm Account Change** dialog box advising **This will cause the service to be restarted**, click **Yes**. (Allow the restart)
21. Back in the **SQL Server (AGINSTANCE) Properties** dialog box, click **OK**.



*If you are not positive the service stopped and restarted:*

- a. *Right-click on SQL Server (AGINSTANCE), then click Stop.*
- b. *Right-click on SQL Server (AGINSTANCE), then click Start.*

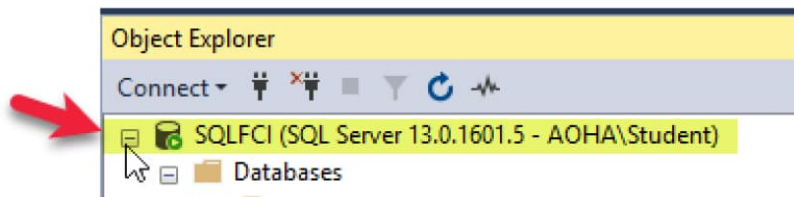
22. Close **SQL Configuration Manager**.

All of the replicas need the ability to log on to the other replicas. This is fairly simple in a domain, but in a non-domain installation you will need to add accounts into each instance of SQL.

23. Using **Hyper-V Manager**, switch to **SQL2** and **SQL3** virtual machines and repeat these steps.

## LAB: SET UP FOR AVAILABILITY GROUPS

1. Using **Hyper-V Manager**, switch to **SQL1** virtual machine (if logon is required, log on as Student).
2. Switch to **SQL Server Management Studio**.
3. Navigate to **Object Explorer**, and collapse the connection into **SQLFCI**.



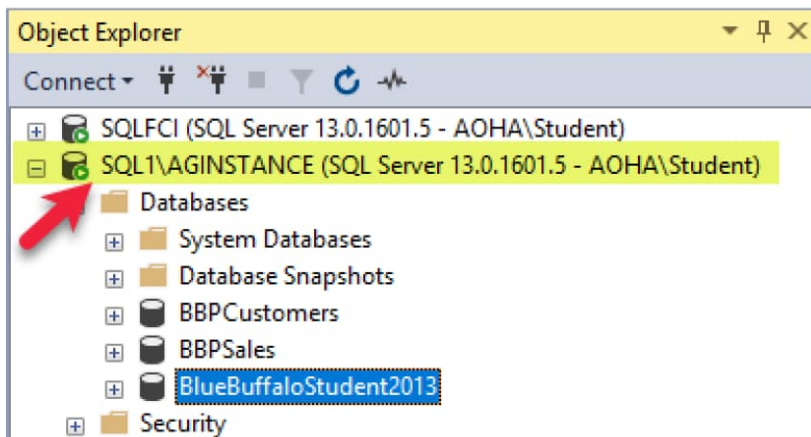
4. Move up to the toolbar within **Object Explorer**, and click **Connect | Database Engine....**
5. In the **Connect to Server** dialog box, move to the **Server name** setting, use the corresponding drop-down arrow and click to select **Browse for more....**

6. In the **Browse for Servers** dialog box, move to the **Select the server to connect to** section and expand **Database Engine**.
7. Click to select **SQL1\AGINSTANCE**.  
*Here because you are on the SQL Server that is hosted on this server you can connect.*  
*When we created the SQLFCI we clustered the Default instance.*
8. Click **OK**.
9. Click **Connect**.
10. Switch to **File Explorer**.
11. Navigate to the C:\DatabaseMDF folder.
12. Once inside the folder, copy the three MDF files:
  - **BBPCustomers**
  - **BBPSales**
  - **BlueBuffaloStudent2013**
13. In **File Explorer** navigate to **C:\Program Files\Microsoft SQL Server\MSSQL13.AGINSTANCE\MSSQL\DATA**.
14. In the **DATA** dialog box advising **You don't currently have permission to access this folder**, click **Continue**.
15. Paste the three mdf files you copied within this folder.
16. Close **File Explorer**.
17. Switch back to **SQL Server Management Studio**.
18. Navigate to **Object Explorer** pane on the left, right-click **Databases** and click **Attach....**
19. Click **Add....**
20. In the **Locate Database Files** dialog box, double-click to select any of the three copied mdf files.
21. Back in the **Attach Databases** dialog box, review the settings, then click **OK**.
22. Navigate to **Object Explorer** pane on the left and expand **Databases** folder.
23. Right-click the database you just attached, then click **Properties**.
24. In the **Database Properties** dialog box, move to the **Select a page** pane on the left and click to select **Options**.
25. Locate the **Recovery model** setting, use the corresponding drop-down arrow and click to select **Full**.
26. Click **OK**.
27. Again, right-click the database you just attached, then click **Tasks | Back Up....**

28. In the **Back Up Database** dialog box, review the settings, then click **OK**.
29. In the **Microsoft SQL Server Management Studio** dialog box advising **The backup of database... completed successfully**, click **OK**.
30. Repeat these steps for the other two remaining databases.

## LAB: THE AVAILABILITY GROUP WIZARD

1. Using **Hyper-V Manager**, switch to **SQL1** virtual machine (if logon is required, log on as Student).
2. Switch to **SQL Server Management Studio** and verify you are still connected to **SQL1\AGINSTANCE**.



3. Expand **Always On High Availability** folder.
4. Right-click **Availability Groups** folder, then click **New Availability Group Wizard....**
5. When the **New Availability Group** dialog box opens, review the **Introduction**, then click **Next**.
6. In the **Specify Availability Group Options** dialog box, locate the **Database Level Health Detection** setting and place a check in the corresponding check box.

This is a SQL 2016 feature that now allows the monitoring of the database itself. Also note the checkbox for DTC which is the support for Distributed Transactions. This feature is enhanced with SQL 2017

<https://docs.microsoft.com/en-us/sql/database-engine/availability-groups/windows/transactions-always-on-availability-and-database-mirroring>

Finally, if you try to pull down the selector in Cluster type nothing else is available. This changes in SQL 2017.

<https://docs.microsoft.com/en-us/sql/database-engine/availability-groups/windows/specify-availability-group-name-page>

7. Navigate to the **Availability group name** text box and enter AGSales.
8. Click **Next**.
9. In the **Select Databases** dialog box, click the **Meets prerequisites** link next to any of the databases listed.

Recall we changed the recovery property to **Full** and did a backup. Else the database would have an indicator here and show you what prerequisites were not met.

10. In the **Microsoft SQL Server Management Studio** dialog box advising **This database meets the availability-database prerequisites**, click **OK**.
11. Place a check in the following two check boxes:
  - **BBPCustomers**
  - **BBPSales**
12. Click **Next**.
13. In **Specify Replicas** dialog box, note that **SQL1\AGINSTANCE** (the server we are on), is selected as **Primary**.
14. Click **Add Replica....**
15. In the **Connect to Server** dialog box, move to the **Server name** text box and enter **SQL2\ AGINSTANCE**.
16. Click **Connect**.
17. Back in **Specify Replicas** dialog box, review the settings.
18. Place a check in the **Synchronous Commit** check box for both replicas.
19. Move to the **Endpoints** tab and review the settings.

Note the account that the endpoint runs under and the name of the endpoint which is the same as it was under mirroring. This name is configurable which has its pros and cons. We will leave it as is.
20. Move to the **Backup Preferences** tab and review the settings.
21. Move to the **Listener** tab and review the settings.
22. Click the radio button to **Create an availability group listener**.
23. Navigate to the **Listener DNS Name** text box and enter **AGL i stener**.
24. Move to the **Port** text box and enter **1433**.
25. Ensure that **Network Mode** is set to **Static IP**.
26. Click **Add....**
27. Navigate to the **IPv4 Address** text box and enter **10.0.0.60** (assuming that is available on the network).



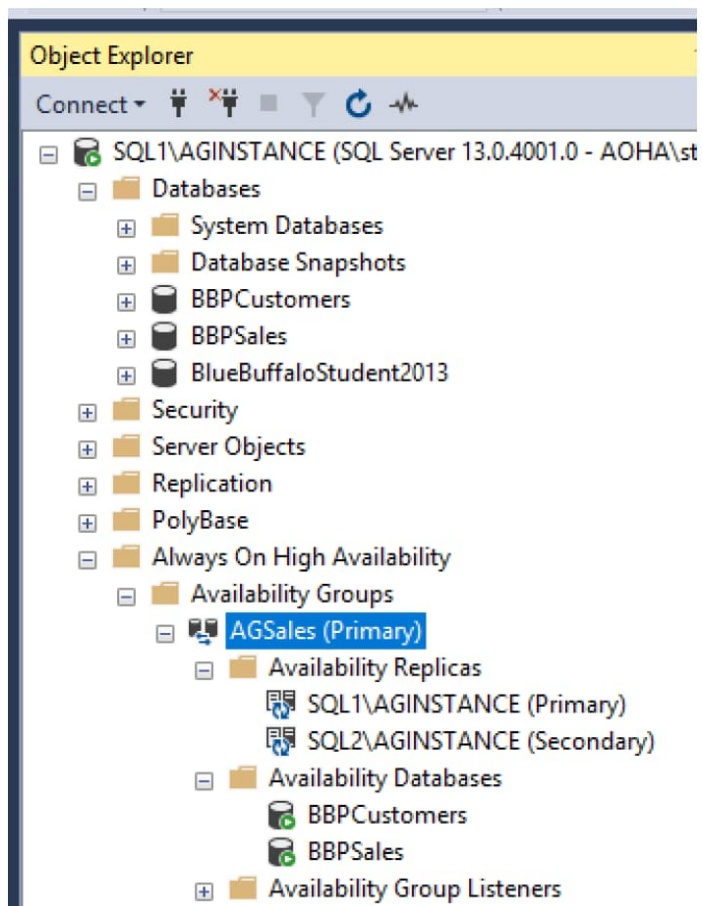
*If unsure about availability, open a CMD prompt and run PING.*

28. Click **OK**.
29. Back in the **Specify Replicas** dialog box, click **Next**.

30. In the **Select Initial Data Synchronization** dialog box, review the options and leave **Automatic seeding** selected.
31. Click **Next**.
32. In the **Validation** dialog box review the results.
33. Click **Next**.
34. In the **Summary** dialog box, click **Finish**.
  - Optional: you can expand Details during the installation.
35. In the **Results** dialog box, review the results, then click **Close**.

## SSMS AND AVAILABILITY GROUPS

(Instructor might show this in her VM.)

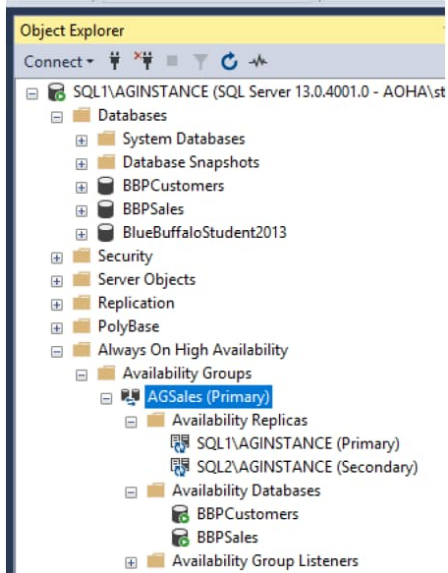


## LAB: SSMS AND AVAILABILITY GROUPS

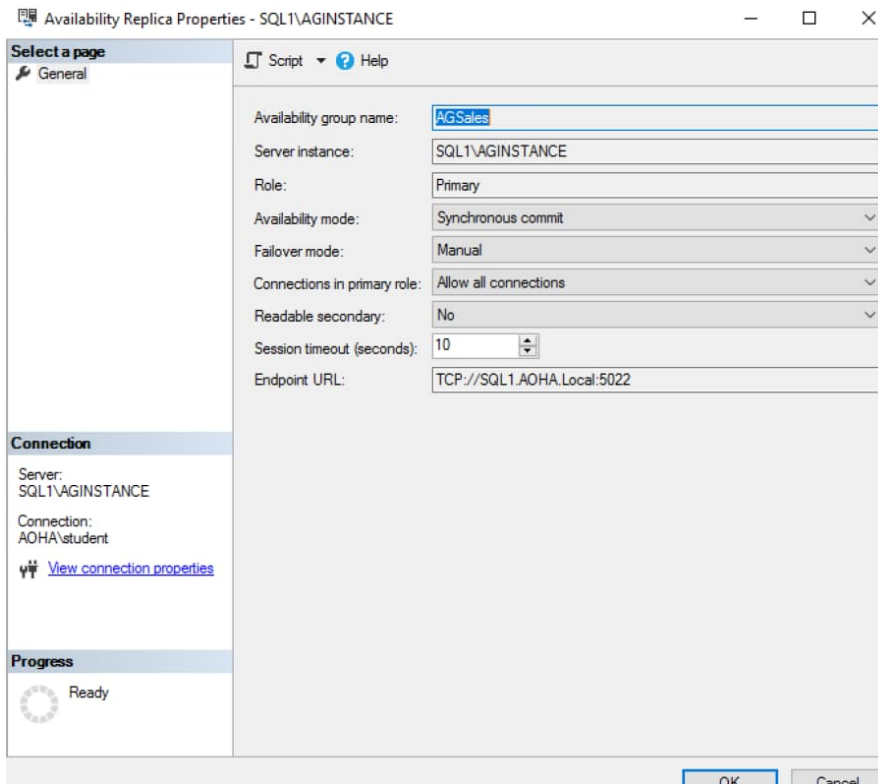
Note to the Instructor: It may be useful here to have the students follow along on their lab environment while you explain. We suggest this as a group exercise encouraging discussion.

1. Switch to SQL1

2. If necessary logon
3. Open SSMS
4. Connect to SQL1\AGInstance
5. Connect to SQL2\AGInstance
6. In SQL1 expand the Availability group and review each section

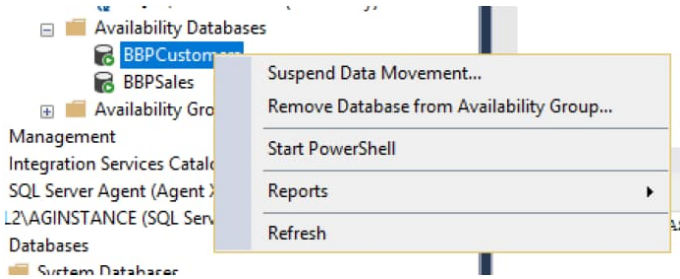


7. Select the AGSales and open Properties



8. Review all of the information on the Properties page noting that this is a summary of what you configured in the wizard

9. Move through the Availability Replicas in SSMS and review the properties page of each noting the differences
10. Move to the databases and right click noting the options.



11. Move to the Availability Group Listener and view the properties
12. In SSMS in Object Explorer expand Always On High Availability
13. Expand Availability Groups and note the differences
14. Move again through the objects looking for any differences.

## THE DASHBOARD

Covered in more depth later in course

Instructor might show this in her VM.

SQL1\AGINSTANCE - ...\_health: event\_file SQLQuery1.sql - S...AOHA\student (54))\* AGSales:SQL1\AGINSTANCE

AGSales: hosted by SQL1\AGINSTANCE (Replica role: Primary)

Availability group state: ✔ Healthy

Primary instance: SQL1\AGINSTANCE

Failover mode: Manual

Cluster state: StudentWindowsCluster (Normal Quorum)

Availability replica:

Name	Role	Failover Mode	Synchronization State	Issues
SQL1\AGINSTANCE	Primary	Manual	Synchronized	
SQL2\AGINSTANCE	Secon...	Manual	Synchronized	

Group by ▾

Name	Replica	Synchronization State	Failover Read...	Issues
<b>SQL1\AGINSTANCE</b>				
BBPCustomers	SQL1\AGINSTANCE	Synchronized	No Data Loss	
BBPSales	SQL1\AGINSTANCE	Synchronized	No Data Loss	
<b>SQL2\AGINSTANCE</b>				
BBPCustomers	SQL2\AGINSTANCE	Synchronized	No Data Loss	
BBPSales	SQL2\AGINSTANCE	Synchronized	No Data Loss	

Last updated: 9/12/2017 10:20:59 AM

Auto refresh: on ||

[Start Failover Wizard](#)

[View Always On Health Events](#)

[View Cluster Quorum Information](#)

[Add/Remove Columns](#)

[Add/Remove Columns](#) ⊞

[Add/Remove Columns](#) ⊞

## LAB: THE DASHBOARD

Always On Dashboard is a Graphical User Interface (GUI) that, similar to your automobile's dashboard, organizes and presents the information about availability groups in a way that is easy to interpret.

Benefits include:

- Health monitoring of replicas and databases
- Administration
- Troubleshooting

Note to the Instructor: It may be useful here to have the students follow along on their lab environment while you explain. We suggest this as a group exercise encouraging discussion.

SQL1\AGINSTANCE - ...\_health: event\_file SQLQuery1.sql - S...AOHA\student (54))\* AGSales:SQL1\AGINSTANCE

AGSales: hosted by SQL1\AGINSTANCE (Replica role: Primary)

Availability group state: ✔ Healthy

Primary instance: SQL1\AGINSTANCE

Failover mode: Manual

Cluster state: StudentWindowsCluster (Normal Quorum)

Availability replica:

Name	Role	Failover Mode	Synchronization State	Issues
<span style="color: green;">✔</span> SQL1\AGINSTANCE	Primary	Manual	Synchronized	
<span style="color: green;">✔</span> SQL2\AGINSTANCE	Secon...	Manual	Synchronized	

Group by ▾

Name	Replica	Synchronization State	Failover Read...	Issues
<span style="color: green;">✔</span> SQL1\AGINSTANCE				
<span style="color: green;">✔</span> BBPCustomers	SQL1\AGINSTANCE	Synchronized	No Data Loss	
<span style="color: green;">✔</span> BBPSales	SQL1\AGINSTANCE	Synchronized	No Data Loss	
<span style="color: green;">✔</span> SQL2\AGINSTANCE				
<span style="color: green;">✔</span> BBPCustomers	SQL2\AGINSTANCE	Synchronized	No Data Loss	
<span style="color: green;">✔</span> BBPSales	SQL2\AGINSTANCE	Synchronized	No Data Loss	

Last updated: 9/12/2017 10:20:59 AM

Auto refresh: on ||

[Start Failover Wizard](#)

[View Always On Health Events](#)

[View Cluster Quorum Information](#)

[Add/Remove Columns](#)

[Add/Remove Columns](#) ⊞

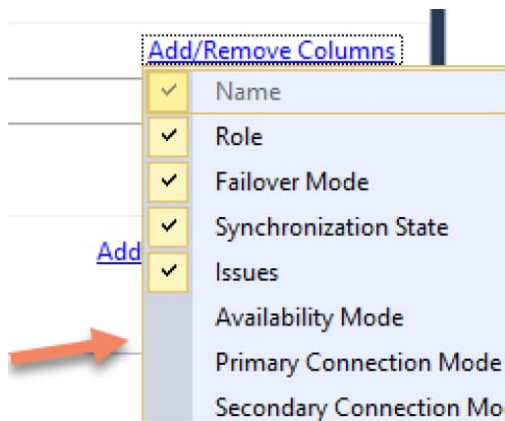
1. On SQL1 in SSMS in Object Explorer right click Availability Groups and select Show Dashboard
2. In object Explorer Move to AGSales right click and Show Dashboard
3. Return to the first tab with the dashboard and expand AGHSales link

Note that the display of the Dashboard is dependent where it was started from

4. Click the link SQL1\AGInstance and note nothing happens
5. Review the information on the Dashboard (We will return to the dashboard later in the course)
6. Move to SQL2\AGINSTANCE link and click it.

Note a new page opens

7. Return to the Dashboard for AGSALES (Primary)
8. There are two Add/Remove Columns
9. Click the top one



10. Place a check next to Availability Mode

Note that it is not uncommon in this interface to not see the added columns because of column sizing. If you do not see Availability Mode look for a faint line right click it and drag it to the left.

Synchronization State	Issues		Synchronization State	Issues	Availability Mode
Synchronized			Synchronized		Synchronous commit
Synchronized			Synchronized		Synchronous commit

Synchronization State	Failover Read...	Issues	Synchronization State	Failover Read...	Issues
Synchronized	No Data Loss		Synchronized	No Data Loss	
Synchronized	No Data Loss		Synchronized	No Data Loss	

Synchronization State	Failover Read...	Issues	Synchronization State	Failover Read...	Issues
Synchronized	No Data Loss		Synchronized	No Data Loss	
Synchronized	No Data Loss		Synchronized	No Data Loss	

11. Move to the lower Add/Remove Columns

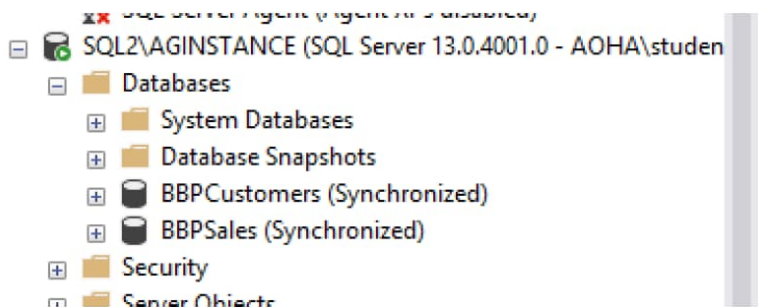
12. Click Add/Remove Columns

13. Place a check in Estimated Recovery Time and Estimated Data Loss

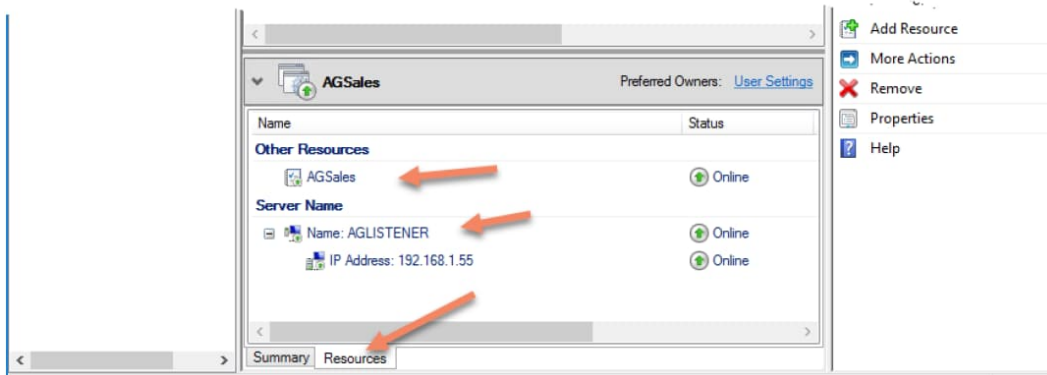
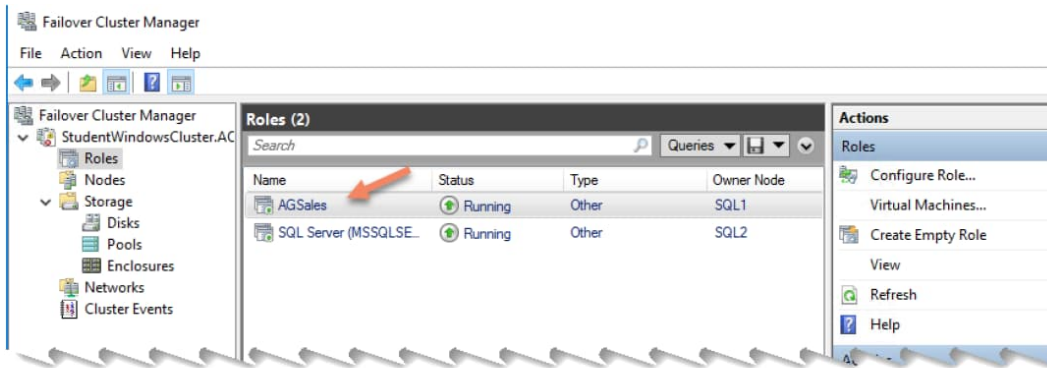
14. Note the changes in the Dashboard. Those are two handy columns though they are complete estimates.

15. In Object Explorer In SQL2 Expand databases

16. Note the status of BBPCustomers and BBPSales



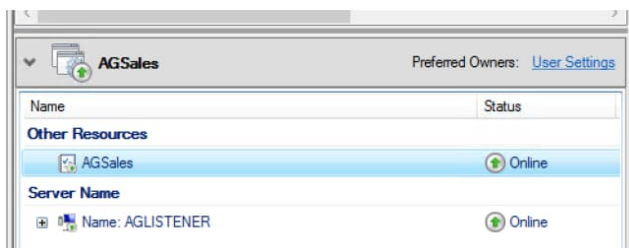
## CLUSTER MANAGER FOR AVAILABILITY GROUPS



1. Open Cluster Manager
2. Move to Roles
3. Select AGSales
4. Move the lower window tab to Resources
5. Note the Listener and its properties

## FLEXIBLE FAILOVER POLICY

1. In Cluster Manager in Roles Click AGSales
2. Under Other Resources Right Click AGSales



3. Select Properties
4. Move to the Properties tab

## AGSales Properties

General Dependencies Policies Advanced Policies Properties

This allows you to view and modify the private properties of this resource.

Name	Type	Value
LeaseTimeout	Read-Write	20000
FailureConditionLevel	Read-Write	3
HealthCheckTimeout	Read-Write	30000
VerboseLogging	Read-Write	0

5. The two you are interested in are FailureConditionLevel and HealthCheckTimeout

The failure condition level and sp\_server\_diagnostics only monitors the health of the SQL Server instance and not the health of the database. However, starting from SQL Server 2016, if you selected Database Level Health Detection while creating the availability group, as we did in the lab, then an automatic failover is initiated when the database status on the primary replica is anything other than ONLINE. If you did not select Database Level Health Detection, then only the health of the instance is used to trigger automatic failover.

See books on line at <https://docs.microsoft.com/en-us/sql/sql-server/failover-clusters/windows/configure-failureconditionlevel-property-settings>

## LAB: REPLICATING LOGINS AND JOBS

In order to log on to a replica we need to replicate the logon and jobs that are stored in Master on the Primary else the secondary is of no value. The next topic of Contained databases is the eventual fix for this issue.

Although there are various methods to perform this the one describe in the Microsoft KB article 'How to transfer logins and passwords between instances of SQL Server'

<https://support.microsoft.com/en-us/help/918992/how-to-transfer-logins-and-passwords-between-instances-of-sql-server>

(Instructor-led students to follow along.)

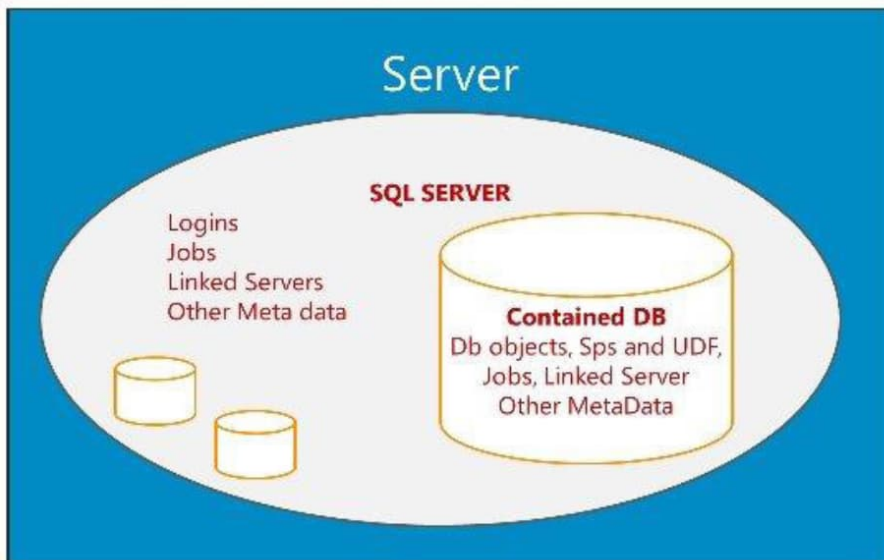
1. On SQL1 open SSMS
2. Connect to both SQL1 and SQL2
3. Open the browser and Move to the Hyperlink above by copying the URL into the address bar
4. Review the script and requirements

5. Copy the script
6. IN SSMS In Object Explorer select SQL1
7. Open a New Query Window
8. Paste the script
9. Execute the script
10. Open a new query window
11. Execute the stored procedure sp\_help\_revlogin as described in the link
12. Copy the output of the Stored Procedure
13. In SSMS move to SQL2
14. Open a new query window making sure it is connected to SQL2
15. Paste the output from the Stored Procedure
16. Execute the script

Note as we created the instance from a script and have not added any logins you will see errors generated.

## CONTAINED OR PARTIALLY CONTAINED DATABASES

### Contained Database



**Note:** This is a feature that is rapidly gaining prominence as it fits well with the concept of Windows Containers or Docker.

SQL Server 2016 supports partially contained databases. A partially contained database allows us to create contained users that authenticate at the database level making the database more independent from the SQL

Server instance.

This feature is further enhanced in SQL 2017

<https://docs.microsoft.com/en-us/sql/relational-databases/databases/contained-databases>