## Abstract

SQL Server 2014 had the capability to store database backups on the Microsoft Azure platform, which brings significant saving in storage cost and minimizes administration effort. The ability to keep database backups on Azure was further enhanced to provide disaster recovery by performing asynchronous replication between the on-premise database and the cloud database backup. This enhancement reduces CAPEX and OPEX cost of physically maintaining additional hardware in the data centers.

In the next version of SQL Server, SQL Server 2016, Microsoft introduces a new feature, called Stretch Database. This feature brings an ability to extend the “on-premise” SQL database to the Azure platform. This feature allows the movement of historical data from on-premise SQL database to an Azure SQL database transparently while allowing access to the data seamlessly. This article explains business use cases, business benefits and technical implementation of the “Stretch Database” feature.

### Use cases for Stretch Database

The following are use cases for Stretch Database feature.

* Archiving of historical data onto low-cost cloud storage to meet the reporting requirements of business users and the compliance requirements of an organization
* Meeting business SLAs with the automated administration of Azure SQL database platform

### Business Benefits

The following are a few business benefits with the implementation of Stretch Database feature.

* Significant cost savings in terms of  database storage as the historical data is moved to low-cost cloud storage
* Minimized database administration efforts
* Improved  user experience with the application as the underlying database queries perform better while accessing the transactional data

### Limitations

The following are limitations with the Stretch Database feature as of SQL Server 2016 CTP Version 2.2.

* Entire data is moved to Azure SQL database table once Stretch database feature is enabled. It is not possible to move specific set of records based on some defined criteria, like WHERE clause.
* Database backup does not include the data from Azure SQL database, but includes only the local data, eligible data at that particular point in time and the remote end point location.

Both of these limitations are expected to be taken care in the SQL Server 2016 RTM version.

## Stretch Database

The Stretch Database feature allows the movement of warm and cold data automatically to the Azure SQL database platform. This feature is beneficial to keep on-line transactional data for longer periods of time inside on-premise storage and move historical data to cloud storage. The query performance is improved while accessing transactional data from the on-premise database as the query optimizer only has to process a limited amount of data. The historical data will be migrated to Azure SQL database transparently and the client applications will be able to query the data seamlessly without requiring any changes to application code.

### Prerequisites

The following are prerequisites to leverage the Stretch Database feature.

* The availability of an Azure SQL database
* An Azure account and subscription for billing. The Azure subscription can be verified at: [https://manage.windowsazure.com](https://manage.windowsazure.com/)
* Setting up of server-level and database-level firewall settings**\*** for the master and the user databases in the Azure SQL database server to selectively allow access to the database.
* Is also recommended you have the *Stretch Database Advisor tool***\*\***

Note:\* - Microsoft recommends using database-level firewall rules wherever possible to make the user database more portable. The server-level firewall rules may be used when there are many databases existing with same access requirements.

Note: \*\* The Stretch Database Advisor is part of SQL Server 2016 Upgrade Advisor. SQL Server 2016 Upgrade Advisor is a standalone tool that enables users of prior versions to run a set of upgrade rules against their SQL Server database to pinpoint breaking and behavior changes and deprecated features as well as providing help with the adoption of new features such as Stretch Database. This tool does not come along with SQL Server 2016 CTP2, can be downloaded from: [here](http://go.microsoft.com/fwlink/?LinkID=613421).

### Implementation

**Step 1:**Set the "remote data archive" instance-level configuration parameter using sp\_configure, as mentioned below.

--To verify the current setting for the instance

EXEC sp\_configure 'remote data archive';

GO

--To set the option to ON at instance level

EXEC sp\_configure 'remote data archive’, '1';

GO

RECONFIGURE;

GO

With this configuration is done, one can enable Stretch Database feature for database, migrate data, and query data on the remote endpoint.

**Step 2:** Enable Stretch feature for the database using either SSMS wizard or T-SQL.

Use can use the SSMS wizard to select the database and provide necessary server location information and login credentials for Azure. This will create a securelinked server definition on the local SQL Server instance to the remote SQL database on cloud.

You will need to create a credential for Azure SQL database using the SSMS wizard (at the time of enabling Stretch Database), or you can use the T-SQL shown below.

-- Create a CREDENTIAL for Azure SQL database server

CREATE CREDENTIAL <Azure\_server\_address>

WITH IDENTITY = <administrator\_user\_name>

, SECRET = <administrator\_password>

Enable Stretch Database feature from master database, as shown below.

ALTER DATABASE <local database name>

SET REMOTE\_DATA\_ARCHIVE = ON (SERVER = <remote server name>);

GO;

**Step 3:** Run the Stretch Database Advisor to identify the candidate tables for Stretch Database enablement. Select all the tables that require stretch enablement in the SSMS wizard, or use the following T-SQL code to enable for each table.

ALTER TABLE dbo.<TableName>

ENABLE REMOTE\_DATA\_ARCHIVE WITH (MIGRATION\_STATE = ON);

GO

### Verification & Validation of Stretch Database Setup

The following are a few checks for the Stretch Database setup.

First, check the value of the column, is\_remote\_data\_archive\_enabled, in the sys.databases and sys.tables system tables (0 – not enabled, 1 – enabled) for each database and each table to know whether the particular database or table is stretch-enabled.

Check the status of data migration to Azure for every database and table by querying the sys.dm\_db\_rda\_migration\_status DMV.

Use the catalog views sys.remote\_data\_archive\_databases and sys.remote\_data\_archive\_tables to list all the remote databases and tables that are used by local Stretch-enabled database.

### Considerations for Stretch Database feature

Stretch enablement does not change the permission model for an existing database. The user logins will be able to access the Azure SQL database through the local instance database only. User logins cannot issue queries directly to the remote endpoint (Azure) through the linked server definition.

As of the CTP version 2.2, SQL Server 2016 supports only shallow backup for a Stretch-enabled database. i.e., the database backup does not include data migrated to the remote server, but contains only local data and eligible data at that point in time when the backup runs, along with the remote endpoint location. The ability to take complete backup is expected to be available with SQL Server 2016 RTM version.

When the local backup has been restored, this operation restores the local data and eligible data to the database. After restore operation is completed, the local database contains local data and eligible data from the point when the backup ran. It is required for the local database instance to connect to the Azure SQL database for data reconciliation. Establish connection to the Azure SQL database by using a stored procedure call, as shown below

EXEC sys.sp\_reauthorize\_remote\_data\_archive @azure\_username, @azure\_password;

GO;

Once the connection is established with the remote SQL database, Stretch Database attempts to reconcile eligible data in the local database with remote data by creating a copy of the remote data on the remote endpoint and linking it with the local database.

Once reconciliation runs, the local database and the remote endpoint are in a consistent state. Then, the previous copy of remote data can be deleted. The data restore process can be represented as below.



**Fig 1:** Data Restore Process on to Stretch-enabled database

### Disable Stretch Database feature for a Table

Here are the steps to follow to disable Stretch Database on a table.

**Step 1:** In order to disable Stretch Database feature for a table, pause data migration on the table. This can be done either by using SSMS (select the table, right-click, Go to Tasks -- Stretch -- Pause) or by using T-SQL, as shown below.

ALTER TABLE <table name>

ENABLE REMOTE\_DATA\_ARCHIVE WITH (MIGRATION\_STATE = OFF);

GO;

**Step 2:** Create a new local table with some arbitrary name in the same database schema as the Stretch-enabled table.

**Step 3:** Copy the data from the Stretch-enabled table into the new table by using an INSERT INTO <new table>… SELECT FROM <Stretch-enabled table> statement.

**Step 4:** Drop the Stretch-enabled table.

**Step 5:** Rename the new table with the same name as the Stretch-enabled table that was dropped in the previous step.

### Disable and Drop Stretch-enabled Database

The Stretch database feature can be disabled for a database only when all the tables with in the database are no longer enabled with the Stretch database feature. This feature can be disabled for a database using SSMS (right-click database, select Tasks -- Disable Database for Stretch) or T-SQL statement, as shown below

ALTER DATABASE <database name>

SET REMOTE\_DATA\_ARCHIVE = OFF;

GO;

The database on local instance can be dropped using DROP DATABASE statement. However, the remote Azure SQL database can be dropped via the Azure Management portal only.

### Limitations imposed by SQL Server to enable Stretch Database feature

The Stretch Database feature comes with following limitations.

* Some table types are not eligible for being Stretch-enabled such as memory optimized tables, replicated tables, change tracking enabled tables, CDC enabled tables and tables containing FILESTREAM data and File Table data
* Tables with columns of data type timestamp, sql\_variant, XML, geometry, CLR-defined types, geography, and “Always Encrypted” columns are not eligible for enabling this feature
* Computed columns cannot be migrated
* Uniqueness is not enforced via UNIQUE constraints and PRIMARY KEY constraints on a Stretch-enabled table
* Constraints such as CHECK, DEFAULT and FK will not be migrated
* XML, full-text, spatial, clustered column-store and indexed views that reference the Stretch-enabled table are not eligible for enabling this feature
* DML operations, INDEX operations cannot be performed on a Stretch-enabled table

### Known issues with Stretch Database Configuration

The following are a few known issues encountered while implementing the Stretch Database feature.

**Firewall settings** - Verify the firewall rules configured for the database in the Azure Management Portal. In case of failure to connect with the Azure SQL database, try some of the following:

* Create a firewall exception on the local computer for TCP port 1433, where SQL Server local instance is installed.
* Verify the IP address used by the computer to connect to Azure SQL Database – this may be incorrectly specified in the Azure Management Portal due to Network address translation (NAT). Mention the modified IP Address as per NAT in the Azure Management portal
* The IP address might be dynamic in nature. Get a static IP address, and then add the IP address to a firewall rule.
* Verify the login credentials used to connect with the Azure SQL database

**Data movement:** Rows from the Stretch-enabled table are not being migrated to Azure SQL database. Verify the following:

* Verify network connectivity between the  local SQL Server computer and the Azure SQL database
* Check firewall settings, as mentioned  above
* Check the status of migration in the DMV *sys.dm\_db\_rda\_migration\_status* for the latest batch. Check the data in the columns *error\_number, error\_state, and error\_severity* values for that particular batch.

**Query performance:** Queries accessing the Stretch-enabled table are running slow after Stretch enablement. Verify the following:

* Determine whether the Azure SQL Database server in a different geographical region than the local server. It is recommended to configure the Azure SQL database server to be in the same geographical region as the local SQL Server to reduce network latency.
* The service tier or the performance level of the Azure SQL database server may not be sufficient to process the SQL queries quickly. Make sure appropriate service tier has been employed for Azure SQL database (Basic vs. Standard vs. Premium) depending on the transaction volume/number of concurrent users/business continuity requirements.
* Verify the performance metrics either in Azure Management portal or catalogue views *sys.resource\_stats* in the master database of local server, or use *sys.dm\_db\_resource\_stats* to monitor more granular data within a smaller timeframe.
* Check that the network conditions have not degraded. Contact network administrator for info about recent issues or outages.

## Conclusion

The “Stretch Database” feature introduced in SQL Server 2016 is a way to extend your on-premise SQL Server database to Azure platform. This feature brings significant cost savings to organizations by keeping cold data on cloud storage and enhanced experience for users while accessing on-line transactional data due to improvement in query performance.

## References

1. Stretch database - <https://msdn.microsoft.com/en-us/library/dn935011.aspx>
2. Enable Stretch Database - <https://msdn.microsoft.com/en-IN/library/mt163698.aspx>
3. Stretch Database Advisor - <https://msdn.microsoft.com/en-us/library/dn935004.aspx>
4. Manage  and troubleshoot Stretch Database - <https://msdn.microsoft.com/en-IN/library/mt484151.aspx>
5. Backup and restore Stretch-enabled databases  - <https://msdn.microsoft.com/en-IN/library/dn934993.aspx>
6. Disable Stretch Database - <https://msdn.microsoft.com/en-IN/library/dn935010.aspx>
7. Pause and resume Stretch Database - <https://msdn.microsoft.com/en-IN/library/dn934998.aspx>
8. How to: Configure firewall settings on SQL Database - <https://azure.microsoft.com/en-us/documentation/articles/sql-database-configure-firewall-settings/#next-steps>
9. Azure SQL Database firewall - <https://azure.microsoft.com/en-us/documentation/articles/sql-database-firewall-configure/>
10. Azure SQL Database Service Tiers and Performance Levels - <https://azure.microsoft.com/en-us/documentation/articles/sql-database-service-tiers/?rnd=1>

|  |
| --- |
| **About the Author**: Suresh Yaram is a Data Architect with 16 years of experience in Data Management space. Suresh works in Data Modeling (ER Modeling/Dimensional Modeling), Data Architecture and Database Design and Development of OLTP Systems and OLAP systems using industry standard tools, technologies and frameworks. His areas of interests include Data Modeling, ETL Process Design, Business Intelligence, Big Data, Data Science etc.,. Currently, Suresh works with Computer Science Corporation (CSC) India Limited, Hyderabad, India as Data Architect. |