



SQL Azure vs. SQL Server

Authors

Dinakar Nethi, Niraj Nagrani

Technical Reviewers

Michael Thomassy, David Robinson

Published

April 2010

Summary

SQL Azure Database is a cloud-based relational database service from Microsoft. SQL Azure provides relational database functionality as a utility service. Cloud-based database solutions such as SQL Azure can provide many benefits, including rapid provisioning, cost-effective scalability, high availability, and reduced management overhead. This paper compares SQL Azure Database with SQL Server in terms of logical administration vs. physical administration, provisioning, Transact-SQL support, data storage, SSIS, along with other features and capabilities.

Copyright

This is a preliminary document and may be changed substantially prior to final commercial release of the software described herein.

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This white paper is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED, OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in, or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

© 2010 Microsoft Corporation. All rights reserved.

Microsoft, ADO.NET Data Services, Cloud Services, Live Services, .NET Services, SharePoint Services, SQL Azure, SQL Azure Database, SQL Server, SQL Server Express, Sync Framework,

Visual Studio, Windows Live, and Windows Server are trademarks of the Microsoft group of companies.

All other trademarks are property of their respective owners.

Contents

- 1
- Similarities and Differences 4
 - Logical Administration vs. Physical Administration 4
 - Provisioning..... 4
 - Transact-SQL Support 5
 - Features and Types 5
- Key Benefits of the Service..... 5
 - Self-Managing 5
 - High Availability 6
 - Scalability 6
 - Familiar Development Model 6
 - Relational Data Model 6

Similarities and Differences

Similar to an instance of SQL Server on your premises, SQL Azure exposes a tabular data stream (TDS) interface for Transact-SQL-based database access. This allows your database applications to use SQL Azure in the same way that they use SQL Server. Since SQL Azure is a service, administration in SQL Azure is slightly different.

Unlike administration for an on-premise instance of SQL Server, SQL Azure abstracts the logical administration from the physical administration; you continue to administer databases, logins, users, and roles, but Microsoft administers and configures the physical hardware such as hard drives, servers, and storage. This approach helps SQL Azure provide a large-scale multi-tenant database service that offers enterprise-class availability, scalability, security, and self-healing.

Since Microsoft handles all of the physical administration, there are some differences between SQL Azure and an on-premise instance of SQL Server in terms of administration, provisioning, Transact-SQL support, programming model, and features. For more information, see [Guidelines and Limitations \(SQL Azure Database\)](#).

Logical Administration vs. Physical Administration

Although SQL Azure plays an active role in managing the physical resources of the database, the DBA plays a very important role in administering SQL Azure-based database applications. Using SQL Azure, DBAs manage schema creation, statistics management, index tuning, query optimization, and security administration (logins, users, roles, etc.). For more information about security administration in SQL Azure, see [Managing Logins and Users in SQL Azure](#).

Database administration in SQL Azure differs most from SQL Server in terms of physical administration. SQL Azure automatically replicates all data to provide high availability. SQL Azure also manages load balancing and, in case of a server failure, transparent fail-over to a healthy machine hosting one of the backup copies of your database.

To provide this level of physical administration, you cannot control the physical resources of SQL Azure. For example, you cannot specify the physical hard drive or file group where a database or index will reside. Because the computer file system is not accessible and all data is automatically replicated, SQL Server backup and restore commands are not applicable to SQL Azure. The SQL Azure service still backs up all databases; however they are not accessible to regular users. This is a feature that may be offered in future.

Provisioning

When preparing an on-premises SQL Server deployment, it may be the role of the DBA or IT department to prepare and configure the required hardware and software. When using SQL Azure, these tasks are performed by the SQL Azure provisioning process.

You can begin provisioning your SQL Azure databases after you create a Windows Azure Platform account. This account allows you to access all the services, such as Windows Azure, AppFabric, and SQL Azure, and is used to set up and manage your subscriptions.

Each SQL Azure subscription is bound to one SQL Azure server within one of the Microsoft data centers. Your SQL Azure server is an abstraction that defines a grouping of databases. To enable load-balancing and high availability, databases associated with your SQL Azure server may reside on separate physical computers within the Microsoft data center. Currently our data centers are located at Chicago (North Central US), San Antonio (South Central US), Dublin (West Europe), Amsterdam (North Europe), Singapore (Southeast Asia) and Hong Kong (East Asia). More data centers will likely be added in future.

For more information about provisioning, see [SQL Azure Provisioning Model](#).

Transact-SQL Support

Many SQL Server Transact-SQL statements have parameters that allow you to specify file groups or physical file paths. These types of parameters are not supported in SQL Azure because they have dependencies on the physical configuration. In such cases, the command is considered partially supported. For more information about Transact-SQL support, see [Transact-SQL Support \(SQL Azure Database\)](#).

Features and Types

SQL Azure does not support all of the features and data types found in SQL Server. Analysis Services, Replication, Reporting Services, and Service Broker are not currently provided as services on the SQL Azure. However, you can connect from on-premises Reporting Server or Analysis Server to SQL Azure and SQL Azure can be used either as a data source or destination.

Because SQL Azure performs the physical administration, any statements and options that attempt to directly manipulate physical resources will be blocked, such as Resource Governor, file group references, and some physical server DDL statements. It is also not possible to set server options and SQL trace flags or use the SQL Server Profiler or the Database Tuning Advisor utilities.

Key Benefits of the Service

The benefits of using SQL Azure are manifold. These include manageability, high availability, scalability, a familiar development model, and a relational data model.

Self-Managing

SQL Azure offers the scale and functionality of an enterprise data center without the administrative overhead that is associated with on-premise instances of SQL Server. This self-managing capability enables organizations to provision data services for applications throughout the enterprise without adding to the support burden of the central IT department or distracting technology-savvy employees from their core tasks in order to maintain a departmental database application.

With SQL Azure, you can provision your data storage in seconds. This reduces the initial costs of data services by enabling you to provision only what you need. When your needs change, you can easily extend your cloud-based data storage to meet those needs.

High Availability

SQL Azure is built on proven Windows Server and SQL Server technologies, and is flexible enough to cope with any variations in usage and load. The service replicates multiple redundant copies of your data to multiple physical servers to maintain data availability and business continuity. In the case of a hardware failure, SQL Azure provides automatic failover to ensure availability for your application.

Scalability

A key advantage of SQL Azure is the ease with which you can scale your solution. As data grows, databases need to either scale up or scale out. Scale up always has a ceiling whereas scale out has no virtual limits. A common scale out technique is data-partitioning. After partitioning your data, the service scales as your data grows. A pay-as-you-grow pricing model makes sure that you only pay for the storage that you use, so that you can also scale down the service when you do not need it.

Familiar Development Model

When developers create on-premise applications that use SQL Server, they use client libraries like ADO.NET, ODBC that use the tabular data stream (TDS) protocol to communicate between client and server. SQL Azure provides the same TDS interface as SQL Server so that you can use the same tools and libraries to build client applications for data that is stored in SQL Azure. For more about TDS, see [Network Protocols and TDS Endpoints](#).

Relational Data Model

SQL Azure will seem very familiar to developers and administrators because data is stored in SQL Azure just like it is stored in SQL Server, by using the familiar relational data model. Conceptually similar to an on-premise instance of SQL Server, a SQL Azure server is logical group of databases that acts as an authorization boundary.

Within each SQL Azure server, you can create multiple databases that have tables, views, stored procedures, indices, and other familiar database objects. This data model makes good use of your existing relational database design and Transact-SQL programming skills, and simplifies the process of migrating existing on-premise database applications to SQL Azure. For more about Transact-SQL and its relationship to SQL Azure, see [Transact-SQL Support \(SQL Azure Database\)](#).

SQL Azure servers and databases are virtual objects that do not correspond to physical servers and databases. By insulating you from the physical implementation, SQL Azure enables you to spend time on your database design and adding value to the business.

Feature	SQL Server (On-premise)	SQL Azure	Mitigation
Data Storage	No size limits as such	<ul style="list-style-type: none"> Web Edition Business Edition Exact size and pricing information can be obtained at Pricing Overview . Note: When you reach the allocated level (1 GB or 10 GB), only SELECTs and DELETEs will be supported. UPDATEs and INSERTs will throw an error.	<ul style="list-style-type: none"> An archival process can be created where older data can be migrated to another database in SQL Azure or on premise. Because of above size constraints, one of the recommendations is to partition the data across databases. Creating multiple databases will allow you take maximum advantage of the computing power of multiple nodes. The biggest value in the Azure model is the elasticity of being able to create as many databases as you need, when your demand peaks and delete/drop the databases as your demand subsides. The biggest challenge is writing the application to scale across multiple databases. Once this is achieved, the logic can be extended to scale across N number of databases.
Edition	<ul style="list-style-type: none"> Express Workgroup Standard Enterprise 	<ul style="list-style-type: none"> Enterprise Edition 	
Connectivity	<ul style="list-style-type: none"> SQL Server Management Studio SQLCMD 	<ul style="list-style-type: none"> SQL Server 2008 R2 Management Studio provides complete connectivity to SQL azure. Prior versions have limited support. SQLCMD 	
Data Migration		<ul style="list-style-type: none"> SQL Server Integration Services, BCP and SqlBulkCopyAPI are supported 	
Authentication	<ul style="list-style-type: none"> SQL Authentication Windows Authentication 	<ul style="list-style-type: none"> SQL Server Authentication only 	Use SQL Server authentication
Schema	No such limitation	SQL Azure does not support heaps. ALL tables must have a clustered index before data can be inserted.	Check all scripts to make sure all table creation scripts include clustered index.
TSQL Supportability		Certain TSQL commands are fully supported; some are partially supported while others are unsupported. <ul style="list-style-type: none"> Supported TSQL: http://msdn.microsoft.com/en-us/library/ee336270.aspx Partially Supported TSQL: http://msdn.microsoft.com/en-us/library/ee336267.aspx 	

Feature	SQL Server (On-premise)	SQL Azure	Mitigation
		<ul style="list-style-type: none"> Unsupported TSQL: http://msdn.microsoft.com/en-us/library/ee336253.aspx 	
“USE” command	Supported	Not supported	USE command is not supported because each of the databases created by the user may not be on the same physical server. So the application has to retrieve data separately from multiple databases and consolidate at the application level.
Transactional Replication	Supported	Not supported	You can use BCP or SSIS to get the data out on-demand into an on premise SQL Server. You can also use the SQL Data Sync tool to keep on-premise SQL Server and SQL Azure in sync.
Log Shipping	Supported	Not supported	
Database Mirroring	Supported	Not supported	
SQL Agent	Supported	Cannot run SQL agent/jobs on SQL Azure	You can run SQL agent on on-premise SQL Server and connect to SQL Azure
Server options	Supported	<ul style="list-style-type: none"> Some system views are supported (http://msdn.microsoft.com/en-us/library/ee336238.aspx) 	The idea is most system level metadata is disabled as it does not make sense in a cloud model to expose server level information
Connection Limitations	N/A	<p>To provide fair usage experience to all tenants on the nodes, connections to service may be closed due to one of the following situations:</p> <ul style="list-style-type: none"> Excessive resource usage Long running queries – (over 5 minutes) Long running single transactions between BEGIN TRAN and END TRAN – (over 5 minutes) Idle Connections – (over 30 minutes) 	
SSIS	Can run SSIS on-premise	Cannot run SSIS in SQL Azure	Run SSIS on site and connect to SQL Azure with ADO.NET provider

References:

- [SQL Azure Portal](#)
- [SQL Azure Developer Center](#)
- [SQL Azure Team Blog](#)