



Platform for Hybrid Cloud

Technical White Paper

Published: September 2013 (updated)

Applies to: SQL Server and Windows Azure

Summary: Cloud computing brings a new paradigm shift in computing and organizations continue to realize the potential cost benefits of running their database applications in the cloud with greater scale and flexibility. Microsoft SQL Server runs very well in the cloud environment—your organization can deploy SQL Server in a private cloud, hybrid cloud, or public cloud easily and can experience familiar tools for development and management. In the public cloud, you can choose to run SQL Server in Windows Azure Virtual Machine or Windows Azure SQL Database. Microsoft SQL Server in a Windows Azure Virtual Machine provides full feature parity with on-premises SQL Server. Windows Azure SQL Database offers a strong foundation for building cloud-based relational database applications that scale out to millions of users.

Copyright

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.

© 2013 Microsoft Corporation. All rights reserved.

Microsoft, Access, Azure, Office 365, SQL Server, Visual Studio, Windows, and Windows Server, are trademarks of the Microsoft group of companies.

All other trademarks are property of their respective owners.

Contents

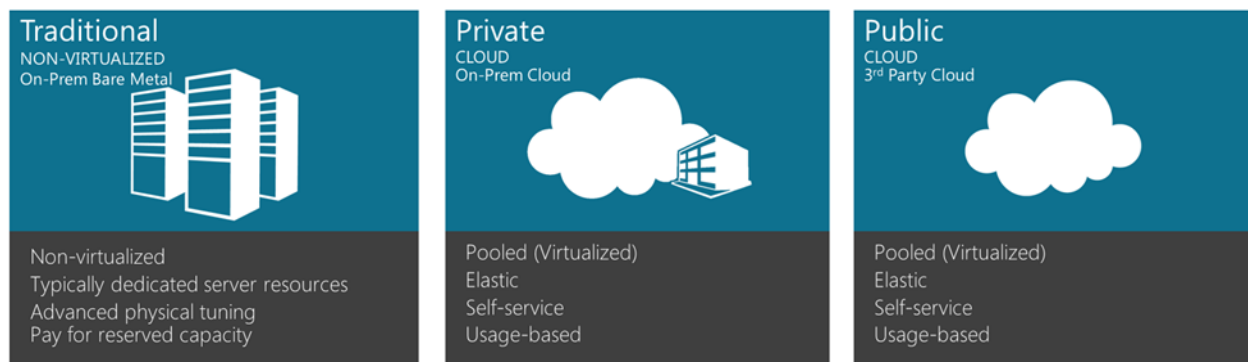
Hybrid cloud: Your database virtually anywhere	4
A portfolio view of database applications.....	4
Traditional bare-metal deployment.....	5
Cloud deployment: Public cloud and private cloud	5
The Microsoft Public Cloud and SQL Server	8
Why and when cloud deployment makes sense	9
SQL Server in a Windows Azure Virtual Machine.....	9
Windows Azure SQL Database	11
Delivering functionality in common cloud scenarios	14
SQL Server in Windows Azure Virtual Machine scenarios	14
Move Existing SQL Server Applications.....	14
Support Development and Test	19
Backup On-Prem Databases to the Cloud	20
Extend On-Prem Applications to Azure Virtual Machine	23
Windows Azure SQL Database scenarios	24
Develop New Cloud-Designed Applications.....	24
Extend On-Prem Applications to Windows Azure SQL Database	27
Hybrid scenarios	28
Code-far apps: A simple path to an enterprise-class data tier	28
Apps occasionally connected with SQL Data Sync	28
SQL Database plus SQL Server in Windows Azure Virtual Machines—in a single app	29
SQL Server Data and Log Files in Windows Azure Storage	30
Conclusion	30
For more information:	31

Hybrid cloud: Your database virtually anywhere

Take almost any application scenario, from the largest public websites to small departmental applications, and you will find the vast majority rely on a database management system of some description. In some respects, developers and IT professionals have become desensitized to the ubiquity of the relational database—it is simply part of the stack for a modern application. As organizations look to take advantage of cloud computing, the availability of cloud-enabled database systems will be critical to their success.

This white paper sets out the Microsoft vision of relational database management systems in the context of cloud computing. It is a hybrid IT vision that gains leverage from the industry-standard Microsoft SQL Server technology set and makes it available across the spectrum of deployment approaches organizations can take today (Figure 1).








Figure 1: Modern IT departments can meet business needs through a combination of on-premises and cloud-hosted delivery



A portfolio view of database applications

The concept of *hybrid cloud* recognizes that organizations typically have a portfolio of different applications deployed across their business and a breadth of environments that have unique requirements. Some applications require detailed and complex hardware configurations that defy deployment into the type of commoditized, “one size fits all” environment offered by cloud computing. Equally, there are workloads in many businesses that are extremely compelling for massive-scale public clouds—it can be economically infeasible to provision sufficient levels of hardware for applications that have massive peaks and troughs in demand. The Microsoft goal for hybrid cloud is to offer organizations breadth of choice in how and where they run their applications, while at the same ensuring they can use a common set of server products, tools, and expertise across a portfolio of solutions (Figure 2).

Figure 2: Each approach to database deployment brings unique benefits and challenges—organizations increasingly are moving workloads to the cloud

	Traditional NON-VIRTUALIZED	Private CLOUD	Public CLOUD
Database Scenarios	Large, Mission-Critical OLTP Data Warehousing Big Data	OLTP BI Workloads Datamarts	OLTP BI Workloads Big Data
Rationale for Usage	 Compliance  Complex, Mission-Critical Applications	 Compliance  Economics Consolidation Cost Allocation	 Rapid Development  Economies of Scale Pay-for-Use  Scalability & Elasticity

Traditional bare-metal deployment

Despite massive improvements in virtualization technology in the past 10 years, the fact remains that there is still a significant performance penalty for virtualizing certain workloads. Large, complex, and mission-critical online transactional processing systems (OLTP) remain the province of massive, dedicated servers that have the operating system and database platform installed directly “on the metal.”

Non-virtualized, dedicated hardware

For most workloads, virtualization is an ideal approach because it delivers significant benefits in total cost of ownership. However, in situations where the ability to scale up matters—where organizations need to extract the most performance possible out of, in many cases, some of the largest server machines money can buy—and where every little bit of extra performance counts, organizations need to run workloads “on the metal.”

A corollary of this requirement to run in a non-virtualized fashion is that applications typically have specific server hardware dedicated to their operation.

Physical tuning

A key benefit of running significant dedicated hardware resources is that there are many opportunities for advanced physical tuning. The most significant area for a database deployment such as SQL Server is the physical configuration of the storage subsystem. The ability to perform physical tuning is something organizations lose when they move to a cloud environment.

Cloud deployment: Public cloud and private cloud

While a private cloud *may* have all the characteristics of a public cloud, that does not necessarily mean it *must* have all of those characteristics. For example, many private clouds do not implement a full charge-back accounting mechanism. Nevertheless, as organizations mature their private cloud strategy, the service and service levels offered by private clouds begin to align more closely with those offered by public cloud providers.

Pooled and virtualized resources

Server virtualization underpins both private and public cloud environments. However, a cloud-based approach to computing requires more than just the mere virtualization of workloads. Many on-premises virtualization environments have specific application targets. Though the environment is virtualized, applications must run on specific, dedicated server hosts. In some cases this is by technical necessity; in others, because a particular department “owns” that node. A cloud environment is predicated on the pooling of hardware resources and while virtualization is a key to pooling capacity, in and of itself it is not enough.

Pooling is the mechanism by which resources are aggregated and then made available as a homogenous pool of capacity capable of running any workload. Workloads that run in a pooled cloud environment are agnostic as to the physical hardware on which they are actually deployed.

Because of the advanced physical tuning required, the Tier-1 workloads discussed above are a pooling anti-pattern. For example, a SQL Server workload that requires a particular approach to physical tuning and certain hard drive spindle layouts could be virtualized but does not lend itself to the use of pooled resources because it has unique resource demands that are unlikely to be demanded by other applications. Put those specific spindle configurations into a pool, and chances are nobody else will want to use them.

Elasticity

Elasticity refers to the ability of the cloud to respond to peaks and troughs in demand. Many business processes have a seasonal characteristic. The agrarian analogy of annual hay-making is an example; most farmers bring in outside contractors with their associated machinery to make hay because it is uneconomic to have the requisite large tractors and hay balers lying idle for most of the year. Information technology workloads also are highly seasonal, yet the machinery deployed to support them is typically purchased in sufficient capacity to meet the peak load and “stored in the shed” for the remaining time.

A typical example of a seasonal workload is the sale of tickets for sporting and cultural events. When a large event goes on sale the number of customers seeking tickets can, in many cases, outstrip supply. Historically, customers would camp all night outside the ticketing office to obtain their tickets. In the online world, this natural queuing mechanism breaks down and instead prospective event-goers swarm the virtual ticketing office, often overloading it.

Because cloud resources are both generic and pooled, it is easy to justify having spare capacity “sitting around.” Cloud providers, whether public or private, typically try to have a portion of their capacity available at all times to deal with peaks. Public clouds are at a distinct advantage here. Because public clouds operate at massive scale, with thousands of customers accessing their pooled resources, they are able to maintain significantly more absolute headroom than a smaller private cloud: one percent of a 100-server cloud does not permit much of a spike in load, whereas one percent of a 10,000-server cloud does. Elasticity is the hardest cloud characteristic to achieve in a private data center because it requires an organization to have capacity lying idle; and avoiding idleness is usually a key justification for cloud-based deployment in the first place.

Some workloads such as the ticketing example above are not feasible in a private cloud environment. A good test of the caliber of a cloud is to ask the question “how many times more capacity does the cloud have deployed than my expected elastic demand?” Capacity should be measured in orders of magnitude and not just mere multiples; if you expect to need tens of servers on a burst basis, then look for a cloud that has at least thousands of nodes.

Self-service

Self-service in cloud computing serves two complementary goals. First, it helps to further drive down the costs of providing the service by reducing or eliminating the labor typically required to provision resources. Second, if done well, it is a measure that benefits users by providing self-service capability. Cloud consumers are empowered to directly access resources. There is no complicated approval process, nor need to wait for the request to become a business priority for IT administrators.

A cloud environment gives users delegated rights to provision resources on demand from the pool. It ensures that users' workloads can't interfere with other workloads and that users may only provision resources up to the capacity level to which they're entitled (or in the case of a public cloud, the limit their credit limit extends to). Self-service drives business agility by allowing organizations to try new things and reach new markets quickly. Whether in a private cloud inside the enterprise or out in Windows Azure, applications can be taken from development to production much more quickly than other through deployment approaches.

Usage-based models

Most shared IT environments suffer from the "tragedy of the commons"¹—if IT capacity is "free" at the margin, then there is no incentive for conservation by any one consumer despite this being in the interest of all consumers collectively. Consumers are used to paying on a per-unit basis for other resources such as water, gas, and electricity. The pay-per-use model offered by cloud computing provides incentives to turn off capacity that is not being used.

Public cloud vendors need to charge for their services, so those environments will always be metered and billed. In private clouds the situation varies; implementing a charge-back model is complex, particularly if the business does not have existing accounting systems in place to support it, but there are significant benefits. The goal of pay-per-use in a private cloud environment is to drive user behavior and to ensure that cloud resources are treated as scarce and conserved where possible. Quotas and other resource allocation mechanisms may be more appropriate for some private cloud environments.

One challenge for public and private cloud operators alike is "which meter to use." What should be counted to determine charges? The metric needs to be well correlated to the actual cost of providing the service but also remain sufficiently simple so that cloud consumers can understand it. It makes little sense to measure "query hours" if cloud consumers do not know how the expected number of required query hours can be derived for their particular application.

The charging model is the mechanism by which a cloud provider signals efficiency. In a cloud environment, where every real resource is obscured by layers of virtualization, organizations should architect their applications to be cost-optimized as a primary consideration.

Compliance

Some application scenarios require compliance with specific enterprise or industry standard policies. These policies typically relate to security, systems management, and legal matters. Policies range from simple things such as which antivirus software to install on servers, to complex Information Security Management Systems standards such as ISO/IEC 27001.

The more control an organization has over the entire systems stack, the more amenable that system is to compliance with all imaginable policies and requirements. An enterprise policy requiring air-gap

¹ "The Tragedy of the Commons." Science 162 (3859): 1243–1248. 1968. <http://www.sciencemag.org/cgi/reprint/162/3859/1243.pdf>

deployment is unsuitable for deployment in a public cloud and, equally, an isolated private cloud cannot coexist on pooled hardware with a private cloud connected to the Internet. The more onerous the compliance requirements, the more likely they are to require a dedicated environment running within the complete control of the enterprise that deploys them.

The major public cloud vendors have moved quickly to audit and certify their systems against industry-standard frameworks. For many small and midsize organizations, achieving compliance can be too costly. For these smaller organizations, public cloud computing actually presents new opportunities to deploy applications into a certified environment without the cost of implementing those standards within their own data centers.

Some compliance challenges remain insurmountable to public cloud computing. Organizations that require complete jurisdiction over their systems often need to ensure that data is located only within a certain country and that systems are accessible only by their own staff. For these organizations, the use of dedicated systems or private cloud environments remains the only feasible solution.

For more information about the compliance of Windows Azure with various industry standards, see <http://www.windowsazure.com/en-us/support/trust-center/compliance/>.

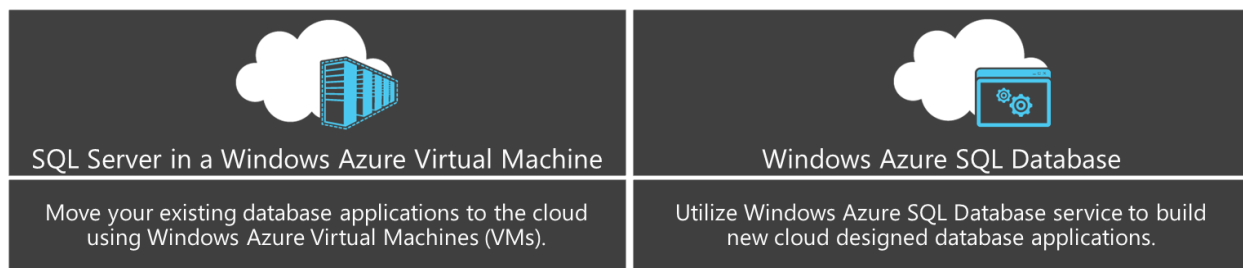
SQL Server, for example, has gone through rigorous compliance exercises since SQL Server 2005 to achieve Common Criteria certification. This certification is formally recognized by the governments of 26 countries that have signed the Common Criteria Mutual Recognition Arrangement (CCRA) and by as many as 40 more governments on a product-by-product basis. The Common Criteria is more than just the concise definitions of security functionalities and assurance requirements. It also is a precise evaluation process defined in the Common Evaluation Methodology document. In addition, it is a formal and approved evaluation scheme for each nation performing Common Criteria evaluations—and it is a government certification based on government working with a private evaluation lab certified in that country. For more information about the compliance of SQL Server, see <http://www.microsoft.com/en-us/sqlserver/common-criteria.aspx#tab1>.

The Microsoft Public Cloud and SQL Server

Windows Azure provides two broad options to enable the use of SQL Server databases:

- **Microsoft SQL Server in a Windows Azure Virtual Machine:** This approach is similar to an on-premises deployment. Organizations have complete control over both the operating system and the installed applications and can achieve almost complete compatibility with on-premises SQL Server. Organizations purchase Windows Azure Virtual Machine time at a few cents per hour with the option to either bring their own SQL Server license or rent the SQL Server license by the hour. SQL Server in a Windows Azure Virtual Machine is ideal for organizations looking to move existing applications to the cloud.
- **Windows Azure SQL Database:** SQL Database is a true platform-as-a-service in the cloud. Organizations purchase this service on a database-by-database basis and Microsoft manages the service all the way to the database level, meaning that organizations do not need to take responsibility for the operating system or even patching the database server software. This service is most suited to organizations building new applications and it provides additional benefits such as high availability and scale-out that are complex to achieve in either the virtual machine approach or on-premises.

Figure 3: The Microsoft public cloud offers two distinct approaches to SQL Server



Why and when cloud deployment makes sense

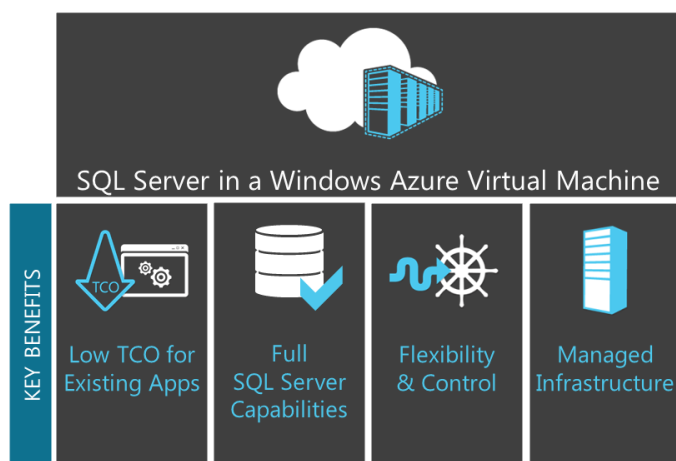
SQL Server in a Windows Azure Virtual Machine

Organizations that use Windows Azure Virtual Machine can create a server in the cloud that they run and manage. These servers can run Windows Server 2008 R2 or later, or a number of different Linux distributions. Organizations have complete control over their server; they can install applications and run almost any workload in the Windows Azure cloud. Installing and running SQL Server in a Windows Azure Virtual Machine is a key scenario that Microsoft delivers and supports.

From a compatibility point of view, running SQL Server in a Windows Azure Virtual Machine is the same as running SQL Server hosted in a virtual machine on-premises. Typically, organizations can move applications to the cloud without making any code changes.

Organizations can create a cloud server by using prebuilt virtual machine images from the Windows Azure Image Gallery (Microsoft provides several images configured with SQL Server Web, Standard, or Enterprise). After they create and start the server, they can move the database onto the server by using one of the common SQL Server database migration techniques—including backup and restore or file detach and attach. Alternatively, organizations can pick up the entire on-premises virtual machine and upload it to Windows Azure.

Figure 4: SQL Server in a Windows Azure Virtual Machine provides an ideal path to move applications to the cloud





SQL Server running in a Windows Azure Virtual Machine offers a low-cost, low-touch migration path for existing apps. The key driver in the total cost of ownership for apps is the cost of developing and maintaining the application itself. In many cases, the operational cost of running inefficient apps in on-premises data centers is still dwarfed by the cost of writing and modifying application code; any cloud migration approach that promises operational savings and requires major code changes is likely to go off track. Moving databases to Windows Azure Virtual Machines typically requires no code changes.

A Windows Azure Virtual Machine running SQL Server Standard Edition (including licenses) costs approximately 64 cents per hour, or about \$476 per month. For workloads suitable for SQL Server Web Edition, costs start at approximately 13.5 cents per hour, or about \$100 per month.

Organizations can achieve even greater cost efficiencies through higher database densities. For an on-premises SQL Server, it is possible to run tens or hundreds of light-load databases on a single virtual machine for a single low monthly cost.



Microsoft SQL Server provides a range of capabilities beyond being a relational database management system (RDBMS). These capabilities include rich reporting technology; extract, transform, and load (ETL) tooling; and job-management and scheduling.

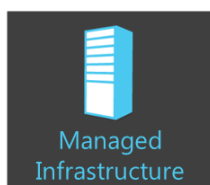
Deploying SQL Server in a Windows Azure Virtual Machine means that organizations can take advantage of the full feature set of whichever version of SQL Server they have deployed. With a couple of minor exceptions, users have access to the complete feature set of the SQL Server boxed product. Features specifically supported include:

- SQL Server Integration Services
- SQL Server Analysis Services
- SQL Server Reporting Services
- SQL Server Master Data Services
- SQL Server Data Quality Services
- SQL Server Agent



By using SQL Server in a Windows Azure Virtual Machine, organizations have complete control over the deployment process. This means that they can customize configuration of Windows and SQL Server. If an application requires the use of third-party tools or technologies, then these items can be installed. Enterprise-level organizations may have specific policies for SQL Server deployments such as password strength requirements or virus scanners; these can be easily installed and enforced in a Windows Azure Virtual Machine.

By using the Windows Azure Virtual Network, IT administrators can domain-join a Windows Azure Virtual Machine to on-premises domains. This capability enables development of hybrid applications that span on-premises and off-premises deployment under a single corporate trust boundary.



Windows Azure provides management of all of infrastructure that underlies the Windows Azure Virtual Machine. IT administrators do not need to manage any of the physical hardware or physical network configuration.

Even so, control comes with increased responsibility. IT departments are responsible for patching the operating system and ensuring that SQL Server is updated.

However, this deployment is just a Windows Server, so they can use standard management technologies, including Microsoft System Center 2012 and earlier versions to help drive this process.

Virtual hard disks (VHDs) are stored in Windows Azure Storage and they benefit from the high-availability storage provided by that service. To achieve a true high-availability database solution, IT administrators must configure the AlwaysOn feature of SQL Server 2012 and later.

Typically, organizations have a 99.9-percent uptime service-level agreement (SLA) for their virtual machine. Organizations are responsible for the uptime of the applications running on that virtual machine, including the SQL Server database service itself.

Windows Azure SQL Database

Windows Azure SQL Database provides a highly available, scalable, multitenant relational database operating in the Windows Azure cloud. SQL Database is ideal for new applications designed for the cloud—these apps can take advantage of its unique scalability features. Because SQL Database is highly compatible with SQL Server, it also is possible to move many existing apps to SQL Database with minimal database and code changes. For organizations that have simple databases they can move, SQL Database offers a very low TCO and benefits that can be hard to achieve on-premises such as high availability and scale-out. It can be a great choice for applications that run entirely in the cloud or, in scenarios where some latency can be tolerated, as the data store for on-premises applications connected remotely to the cloud.

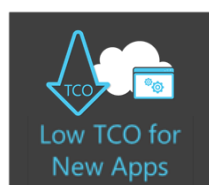
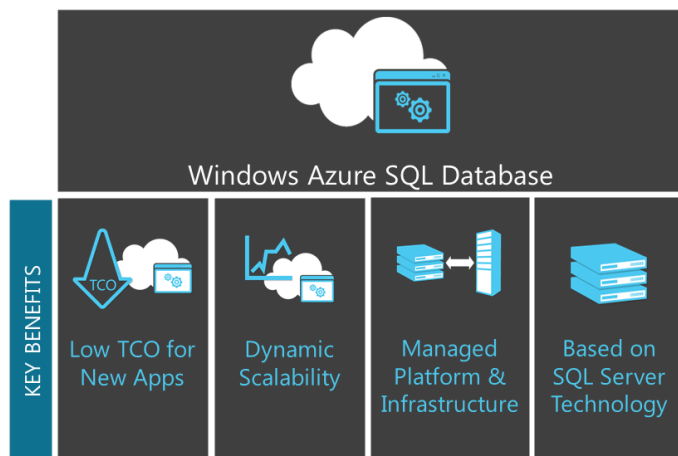
SQL Database has a number of unique features that are not available in cloud offerings from other vendors. While these features work well with the always-on, massive-scale applications currently in demand, they also require specific support from application developers—so they are most suitable for organizations that can take the option to build new apps rather than move existing apps.

The first of these is a feature called SQL Federation. *Federation* enables an organization to scale out a database by providing tooling and T-SQL support for the database *sharding* pattern (row-based partition). Given an appropriately architected application, SQL Database can achieve almost limitless scale in terms of both data volumes and transactional load. To take advantage of this feature requires a specific approach to both database and application design. At a minimum, Federation requires significant rework of an existing application; ideally application developers will build an application for the sharding pattern from the outset.

Another unique feature of SQL Database is that it provides high levels availability by maintaining three transaction-consistent copies of the database distributed across fault-tolerant regions of a Windows Azure data center. If one replica fails, SQL Database reroutes traffic to the remaining operational copies, creates a new replica, and then brings that replica back to transactional consistency. While organizations can use sophisticated high-availability features on-premises and in IaaS deployments, the complexity of setting them up might put them beyond reach. Even the smallest databases, just a few dollars per month in SQL Database, are configured for high availability. To take full advantage of high availability, developers need to ensure that their code can reliably deal with database disconnections; SQL Database will disconnect an application's connections if the workload needs to be moved to another replica. Developers can handle these disconnection events by using retry logic.²

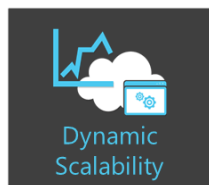
² "Retry Logic for Transient Failures in Windows Azure SQL Database." Microsoft TechNet.
<http://social.technet.microsoft.com/wiki/contents/articles/4235.retry-logic-for-transient-failures-in-windows-azure-sql-database.aspx>

Figure 5: Windows Azure SQL Database provides unique capabilities for building massive-scale applications and offers low TCO for new apps designed for the cloud



By using SQL Database, organizations can deploy their database applications quickly. Because there is no data center to build, hardware to provision, or software to install, a new database can be provisioned in minutes. This agility allows organizations to try new things and reach new markets quickly. Given an appropriate application architecture, the effort to deploy an application for hundreds of users versus hundreds of thousands of users is the same—just a few minutes.

SQL Database is broadly equivalent to an on-premises instance of SQL Server running with full hot standby servers; not one standby server, but two. Such a solution on-premises would typically require thousands of dollars of investment in hardware and setup time, whereas SQL Database starts at just a few dollars per month. This bears repeating: *SQL Database provides a true high-availability, enterprise-class relational database starting at US \$5 per month.*



In the completely connected world of the Internet, phenomenal success can be as hard to deal with as object failure. If a newly launched business application takes off, maybe thanks to good press on one of the heavily trafficked blog sites, it needs to scale fast. “Fast” in this context means within minutes or hours; if it takes days or weeks to add more capacity, then you may have missed the moment.

To achieve rapid scale-out, a deployment platform needs two key properties:

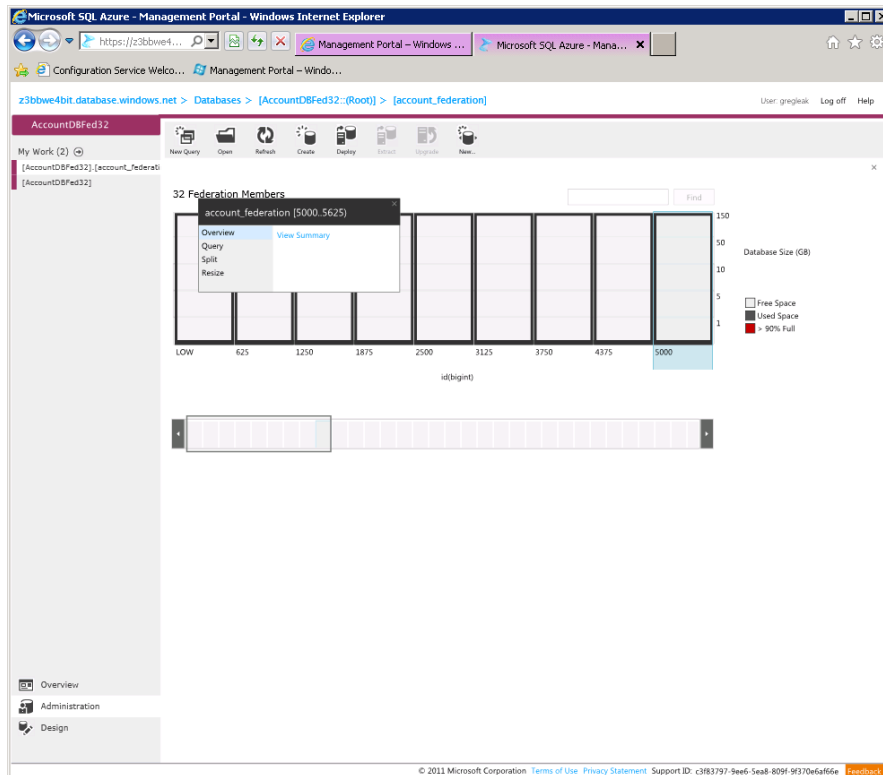
- An **ability to add capacity** without affecting the operational system
- A strong commitment that there will be **sufficient resources** available regardless of how quickly demand grows

By using SQL Federation, developers who build applications with SQL Database can shard their data out across multiple databases. This scale-out happens quickly, is transparent to users, and does not need significant amounts of input from IT professionals. Databases can be scaled out or split while under load. Split operations (re-sharding) are performed online without application downtime.

Because SQL Database is delivered at massive scale, there is always “spare capacity” available in real time (Figure 6). This availability contrasts with on-premises deployment approaches, even private clouds, where a significant new demand for capacity typically requires new hardware to be deployed into the data center first. A very successful new application can outrun an organization’s ability to procure new server

hardware through their supply chain. It takes days or even weeks to order and provision new on-premises servers.

Figure 6: Organizations can manage federation by using T-SQL, SQL Server Management Studio, or the Management Portal



With SQL Federation, organizations can scale out their *relational* databases; this capability is important for organizations that value strong transactional consistency and other attributes that are typically sacrificed by “NoSQL”-style cloud data stores.



Very few businesses differentiate themselves on how well they manage their IT infrastructure. They want to do a great job; they’ll want to be as good as or better than their competitors, but good management typically is not the source of their strategic advantage. With SQL Database, organizations can take advantage of not only the *economies* of scale but also *qualities* of scale.

SQL Database delivers 99.9-percent availability at the database level, backed by SLA. This means that Microsoft guarantees the uptime of not only the servers that run SQL Database, but also the database itself.

Because it manages both infrastructure and platform components, SQL Database delivers zero downtime updates (hardware upgrades, software patching, and more) with no intervention required of IT administrators. By adopting SQL Database, organizations can deliver their applications that have enterprise-grade availability without incurring enterprise levels of cost.

By spending less time operating existing systems, IT teams are free to spend more time innovating systems and improving applications. The IT focus moves from operations to adding strategic value to the business.



The SQL Database protocol is compatible with SQL Server, meaning that developers and IT pros can use familiar tools such as the Microsoft Visual Studio integrated development environment (IDE) and SQL Server Management Studio. When they work with on-premises applications that use SQL Server, developers and IT pros use client libraries that implement the tabular data stream (TDS) protocol to communicate between client and server. Windows Azure SQL Database provides the same TDS

interface as SQL Server, meaning that those same libraries can be used by applications working with data that is stored in Windows Azure SQL Database. Common approaches to relational database programming include ADO.NET, Entity Framework, and ODBC. SQL Database also supports other platforms with drivers for JDBC and even PHP and Node.JS. Code developed for on-premises SQL Server applications usually can be moved easily to Windows Azure SQL Database.

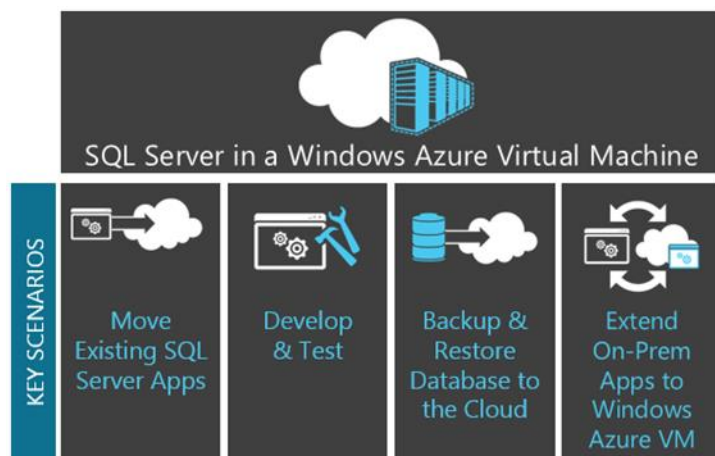
SQL Database is a true RDBMS. The relational database model and the transactional properties of RDBMSs remain the gold standard for enterprise application data storage. Unlike with many other cloud-hosted data stores, developers working with SQL Database do not need to learn new approaches to data-modeling nor make concessions for functionality such as transactional consistency. SQL Database supports stored procedures, stored functions, and the vast majority of tried and true SQL Server features.

Delivering functionality in common cloud scenarios

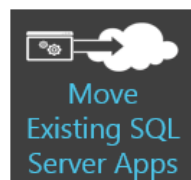
SQL Server in Windows Azure Virtual Machine scenarios

This section covers four key scenarios for organizations that use SQL Server in a Windows Azure Virtual machine. Figure 7 summarizes these scenarios.

Figure 7: SQL Server in a Windows Azure Virtual Machine targets four key scenarios



Move Existing SQL Server Applications

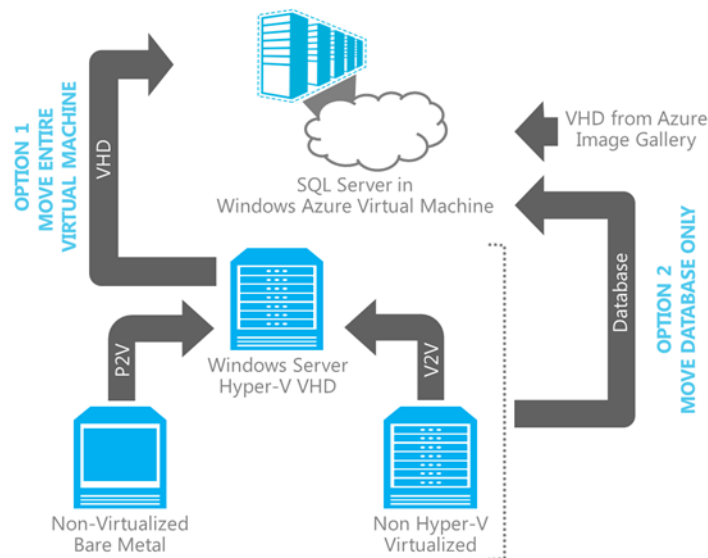


Because it can run nearly any image based on Windows Server, a Windows Azure Virtual Machine facilitates moving an existing application to the cloud. Both the database tier and application tier can be moved to Windows Azure Virtual Machines, or, for some workloads such as ASP.NET, it may make sense to port to Windows Azure Cloud Services or even Windows Azure Web Sites. This platform as a service

(PaaS) application tier can then connect more easily to infrastructure as a service (IaaS) SQL Server running in a Windows Azure Virtual Machine.

Moving a database from an on-premises SQL Server to SQL Server running in a Windows Azure Virtual Machine typically takes one of the paths shown in Figure 8.

Figure 8: Paths for moving a database to SQL Server in a Windows Azure Virtual Machine



Moving the entire virtual machine

In Figure 8, option 1 involves moving the entire virtual machine from an on-premises Hyper-V server to Windows Azure. Moving the entire virtual machine is suited to development and test scenarios, particularly for situations where the on-premises machine has additional and complex configuration. By moving the entire virtual machine, there is no need to re-create that exact configuration in another machine. To move the whole virtual machine, IT pros need to ensure that the server operating system is Windows Server 2008 R2 SP1 x64 or later and that the version of SQL Server is a 64-bit installation of SQL Server 2008 SP3 or later. Organizations with virtual machines that don't meet those specific requirements need to either upgrade or follow the "backup and restore" approach in option 2.

It is also possible to move virtual machines running in a third-party hypervisor such as VMWare, or even physical bare-metal servers, by first moving these server installations into Hyper-V. For this move, organizations must perform either a virtual-to-virtual (V2V) or physical-to-virtual (P2V) migration. Microsoft provides tools in System Center Virtual Machine Manager and Virtual Machine Converter Solution Accelerator.

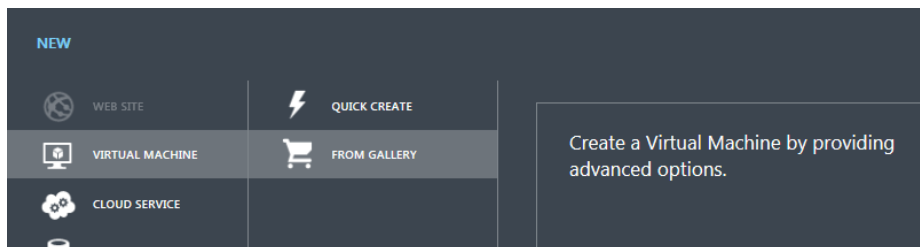
Organizations can upload the virtual machine by using upload tools supported by Windows Azure. System Center App Controller 2012 SP1 also provides a mechanism to move the entire SQL Server virtual machine to Windows Azure.

Moving just the database

In Figure 8, option 2 involves transferring just the database to the cloud. This transfer can take the form of a full database backup or data file, or it may be more appropriate to move the database schema and data separately. A key benefit of this approach is that organizations only need to send the database over the wire; it can be a time-consuming process to send the entire virtual machine on a slow Internet connection.

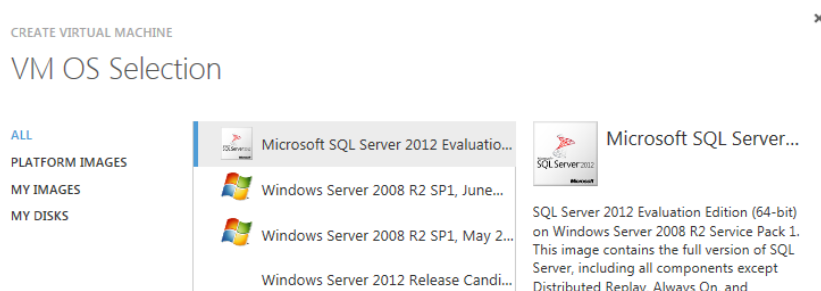
Organizations must start by creating a Windows Azure Virtual Machine from an image in the Image Gallery (Figure 9).

Figure 9: Creating a virtual machine from the Image Gallery



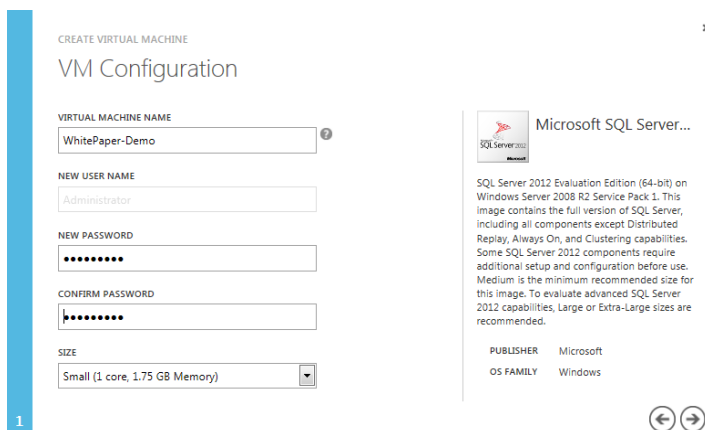
We recommend choosing one of the Microsoft SQL Server images, as shown in Figure 10.

Figure 10: Choosing the Microsoft SQL Server 2012 Evaluation machine image



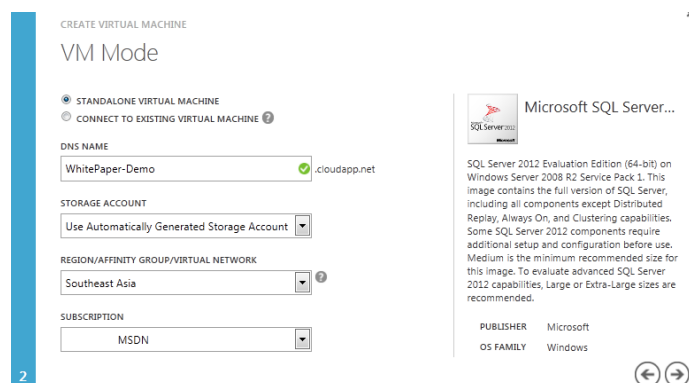
During setup, you must specify a machine name, changing the administrator account name, provide a strong password for it, and choose a machine size. The machine size can be changed later. For testing purposes usually it is sufficient to run a *Small* instance size (Figure 11).

Figure 11: Setting virtual machine properties



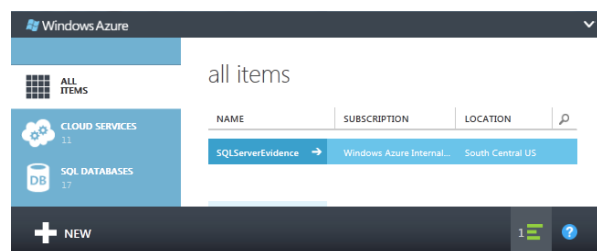
The remaining settings determine where the virtual machine is located, how it is resolved for DNS purposes, and the subscription that will be used (Figure 12).

Figure 12: Setting networking, DNS, storage, data center, and subscription properties



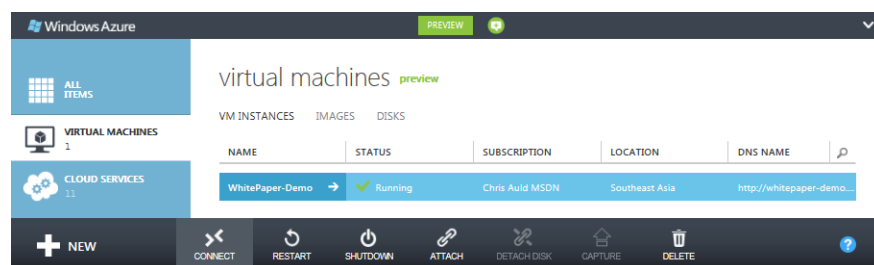
Clicking the ✓ button starts the process of creating the virtual machine, which may take a few minutes. The indicator in the lower-right corner of the portal shows progress in green bars (Figure 13).

Figure 13: Progress indicator for virtual machine creation



In the list of virtual machines in the portal, you must wait for the provisioning process to complete, as indicated in the Status column. You can make a connection after the status is **Running**.

Figure 14: Clicking connect establishes an RDP connection to the virtual machine



At this point, possible approaches include:

- Extract a copy of the on-premises database
- Transfer that copy to the virtual machine running in the cloud

For more information, see “Migrating with SQL Server in Windows Azure Virtual Machines” at <http://msdn.microsoft.com/en-us/library/windowsazure/jj156165.aspx>.

As the steps below show, you can back up the on-premises database (Figure 15), copy the backup file to the server directly over the RDP connection, and then restore the database file to the SQL Server instance running in the Windows Azure Virtual Machine. This approach is well suited to small databases.

The screenshot shows the SQL Server Enterprise Manager interface. In the left-hand tree view, the 'DemoDB' database is selected. A right-click context menu is open, displaying various database management options. The 'Back Up...' option is highlighted in yellow. Other visible options include 'Detach...', 'Take Offline', 'Bring Online', 'Shrink', 'Policies', 'Facets', 'Start PowerShell', 'Launch Database Mirroring Monitor...', 'New Database...', 'New Query', 'Script Database as...', and 'Tasks'.

Figure 16: Copying the file from an on-premises machine and pasting it into the Windows Azure Virtual Machine (Remote Desktop protocol)

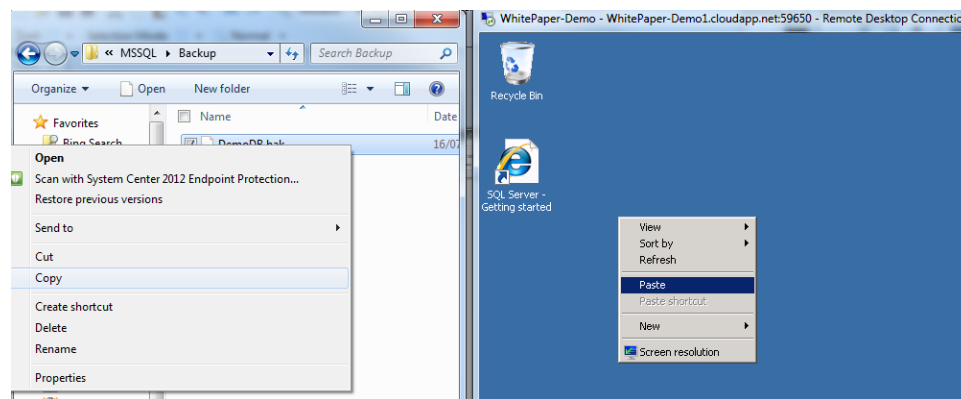
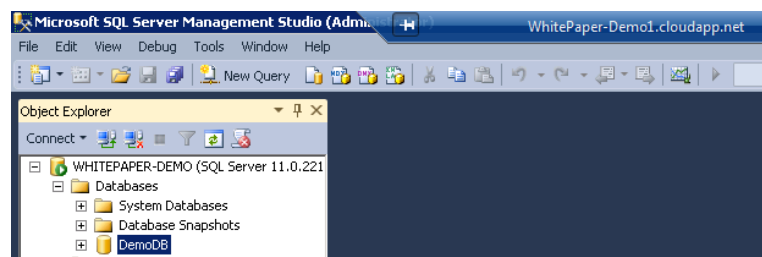


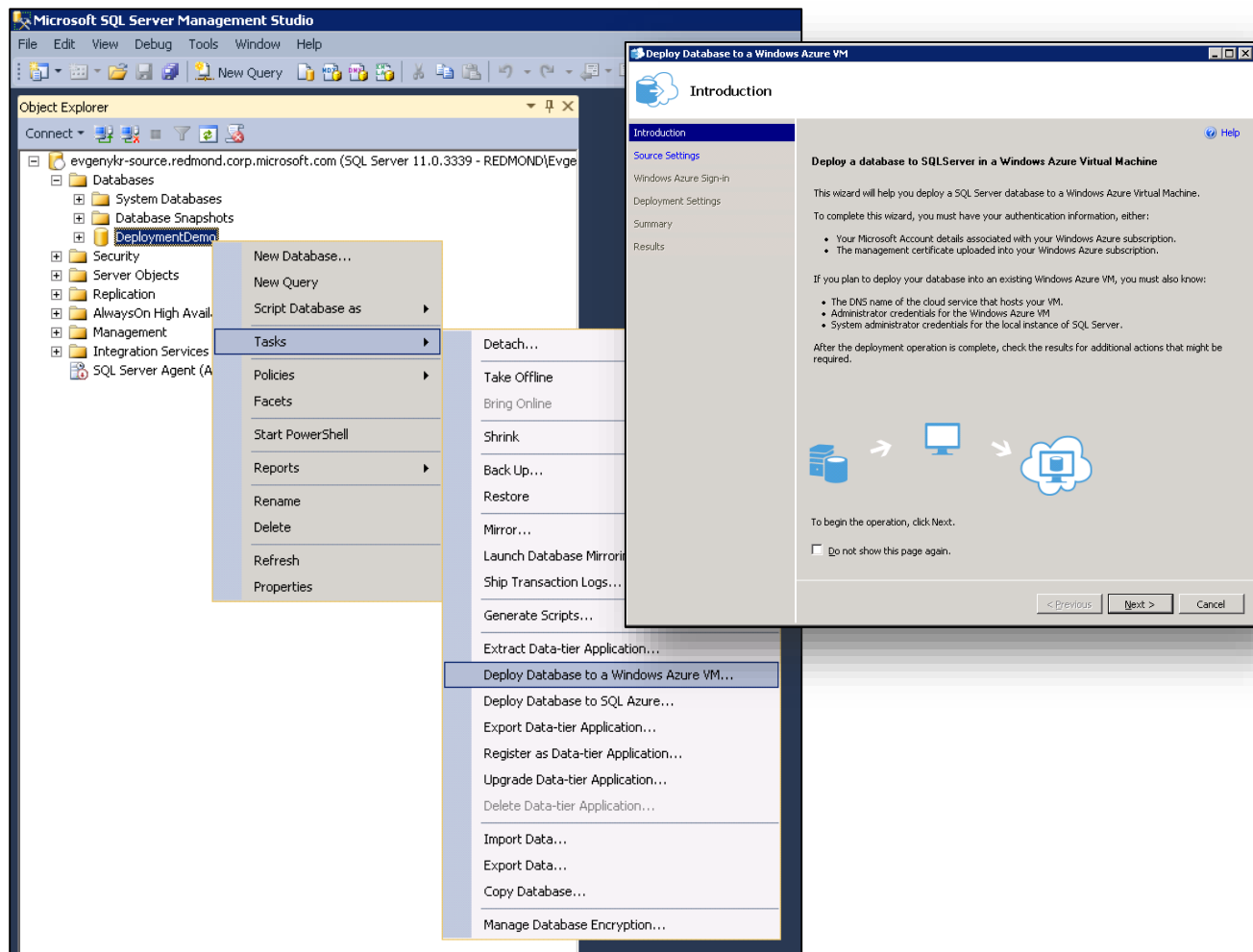
Figure 17: Database restored in SQL Server Management Studio



Page 18

knowledge is not required. There also is no performance overhead for this operation as the major factor defining deployment time is database size. To further simplified the steps of migration, the same wizard can also create the SQL Server instance in a Windows Azure Virtual Machine if needed.

Figure 18: Wizard to deploy on-prem SQL Server database to SQL Server running in Windows Azure Virtual Machine (SQL Server Management Studio)



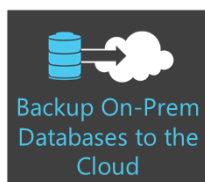
Support Development and Test



When building apps, many organizations use development, test, and staging servers as part of their application life cycle. These setups can range from a virtual machine hosted in a developer's desktop to an environment hosted on dedicated virtual servers. In large organizations, particularly large enterprises and systems integrators, these development servers can require something more akin to a private cloud.

Windows Azure Virtual Machine provides a public cloud option to support these workloads. Even where an application eventually will be hosted on-premises, the use of Windows Azure to run development and staging servers can be a cost-effective choice. It provides the flexibility of self-service machine provisioning and cost-saving benefits of pay-for-use billing.

Backup On-Prem Databases to the Cloud



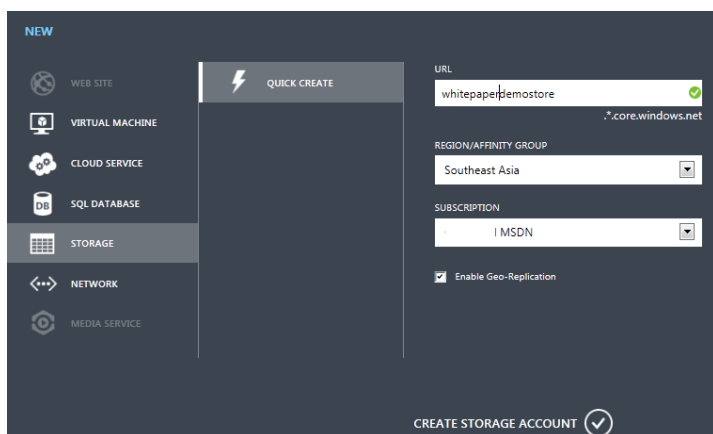
Windows Azure Storage provides a highly available cloud-hosted storage mechanism at a few cents per gigabyte per month. This, coupled with availability of the Windows Azure Virtual Machine service, presents new options for organizations that need backup and disaster recovery capabilities.

Organizations can use Storage to store backups of either their databases or entire virtual machines. Windows Azure Storage provides highly available, durable, and secure offsite storage for these files; three consistent copies are stored in the primary data center and a fourth replica is copied asynchronously to a secondary data center.

Should disaster strike, organizations can either retrieve files from Storage or use a Windows Azure Virtual Machine to restore backed-up machine images and then run them from the cloud for a period.

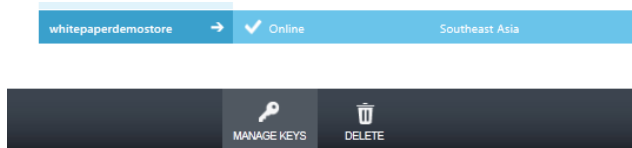
SQL Server 2012 SP1 CU2 onwards includes functionality to enable the use of Azure Storage for database backup and restore (Figure 19).

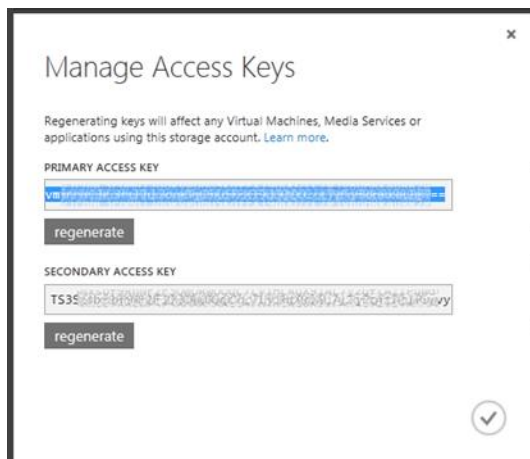
Figure 19: Creating a Windows Azure Storage account using the portal



To upload backup data directly to Windows Azure Storage, you need to retrieve one of the security keys for their storage account (Figure 20).

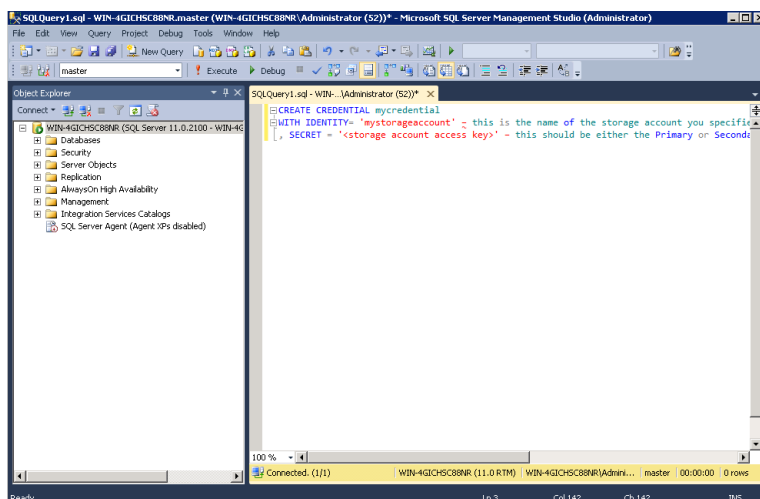
Figure 20: Retrieving storage keys from the portal





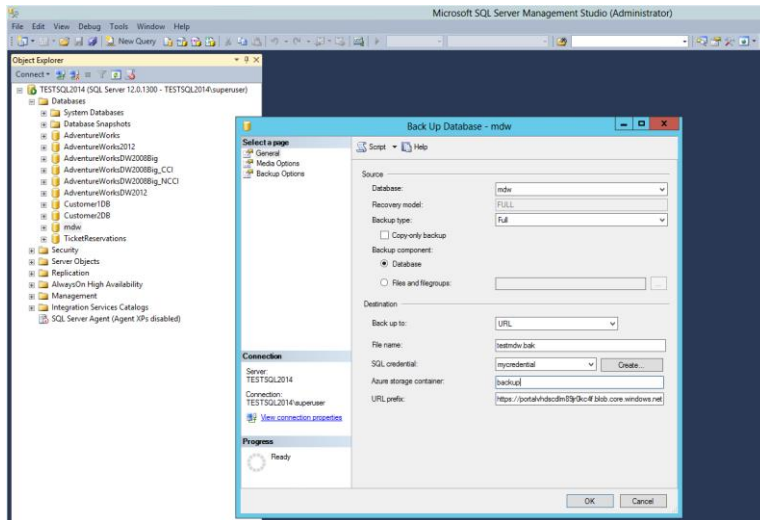
You can gain access to Windows Azure Storage backup support from SQL Server Management Studio and by using the T-SQL command `BACKUP DATABASE TO URL`. First, you need a SQL Server credential (Figure 21) to store authentication information to Windows Azure Storage. The credential stores the name of the storage account and the storage account access key values.

Figure 21: Creating a SQL Server credential to store Windows Azure Storage authentication information



After creating the credential, use `BACKUP/RESTORE` and `WITH CREDENTIAL` to back up and restore the database directly to Windows Azure Storage using SQL Server Management Studio (Figure 22).

Figure 22: Backing up data to Windows Azure Storage in SQL Server 2014 Management Studio

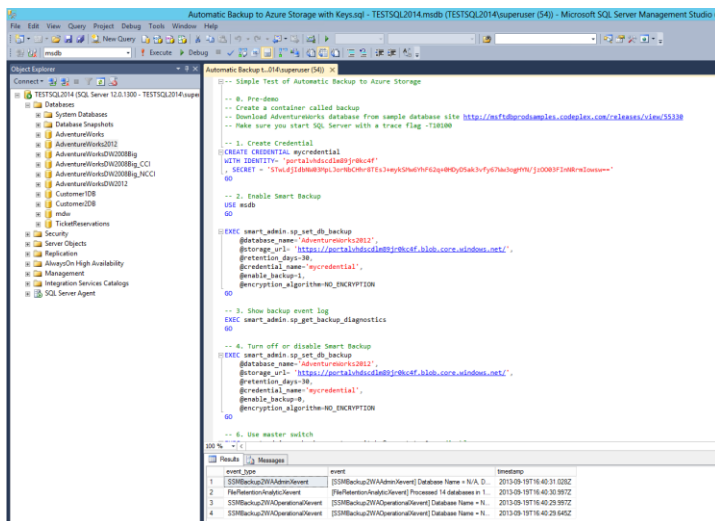


For more information, see “SQL Server Backup and Restore with Windows Azure Blob Storage Service” at <http://msdn.microsoft.com/en-us/library/jj919148.aspx>.

Organizations can use this feature to back up SQL Server databases in an on-premises instance or in an instance of SQL Server running a hosted environment such as Windows Azure Virtual Machine. Backup to the cloud offers benefits such as availability, virtually limitless geographically replicated offsite storage, and ease of moving data to and from the cloud. These benefits also include flexible, reliable, and virtually limitless offsite storage while providing a backup archive mechanism and no overhead for hardware management.

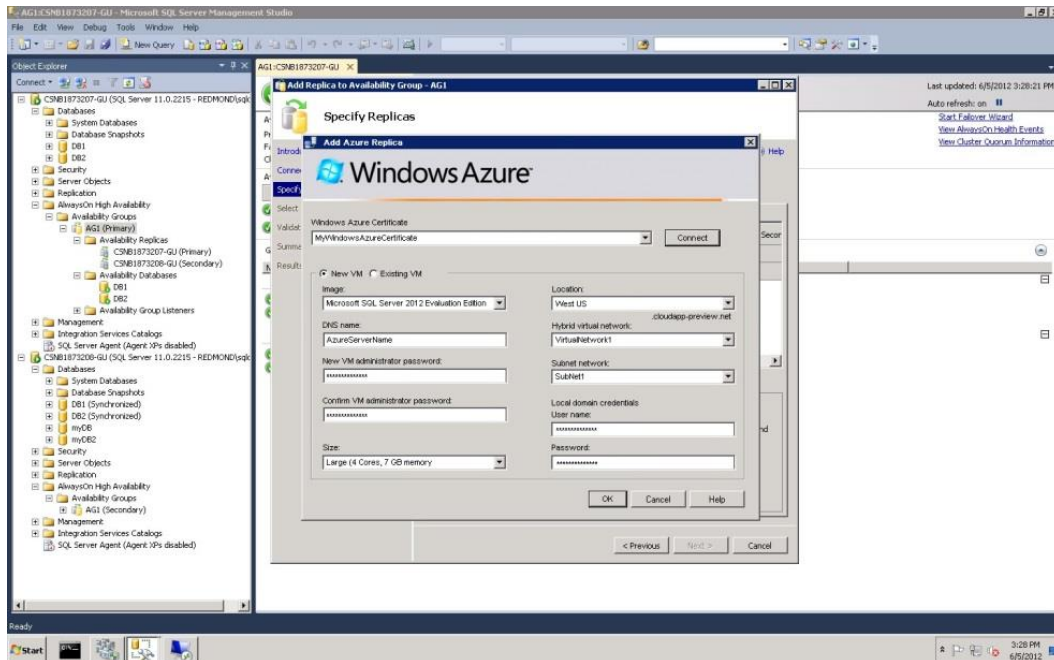
SQL Server 2014 provides cloud backup feature (Figure 23) to automatically backup SQL Server databases to Windows Azure Storage called Managed Backup to Windows Azure. The backup process is context-aware (sensitive to workload and throttling), uses minimum configurations to specify the retention period, and can manage the entire database instance or particular databases.

Figure 23: Managed Backup to Windows Azure in SQL Server 2014 to automatically backup SQL Server databases



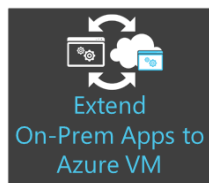
SQL Server 2014 also allows creation of AlwaysOn replica in Windows Azure Virtual Machine for Disaster Recovery. An easy to use wizard (Figure 24) is supplied to make this process seamless. Using this wizard, customers can easily deploy one or more replicas of their databases to Windows Azure Virtual Machine. The benefits are great as customers do not need to set up a DR site, there is no need to purchase DR machines for replicas. After setup, customers can backup their databases using the replica or use the replica for reporting applications.

Figure 24: Wizard to create AlwaysOn replica in Windows Azure Virtual Machine (SQL Server Management Studio)



Finally, in addition to backing up data from within SQL Server, users of Windows Server 2012 and System Center 2012 Data Protection Manager can use Windows Azure Online Backup to back up other data—including whole virtual machine images—directly to Windows Azure and stores all the backup data in a central location.

Extend On-Prem Applications to Azure Virtual Machine



In some situations, organizations may want to move just a portion of an application to the cloud and retain sensitive data on-premises while moving portions of the application that require additional scale to the public cloud.

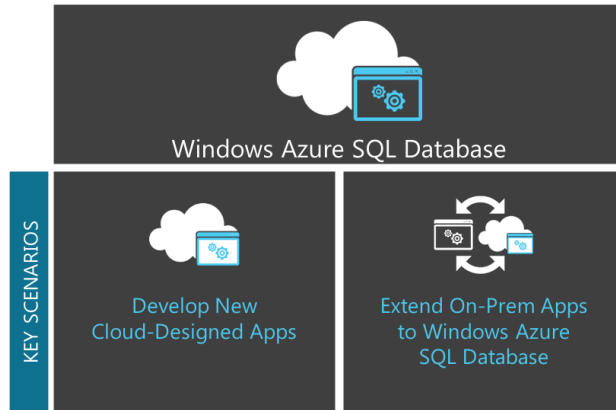
Consider the e-commerce site. Sensitive information such as customer details and credit card information may need to remain on-premises; but, to support increased load, the product catalog and shopping cart data could be moved into Windows Azure.

We discuss these types of hybrid applications in more detail in later sections.

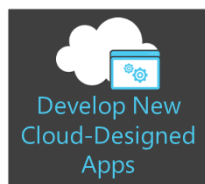
Windows Azure SQL Database scenarios

This section covers two key scenarios for organizations looking to work with Windows Azure SQL Database.

Figure 25: SQL Database is ideal for developers creating apps designed for the cloud



Develop New Cloud-Designed Applications



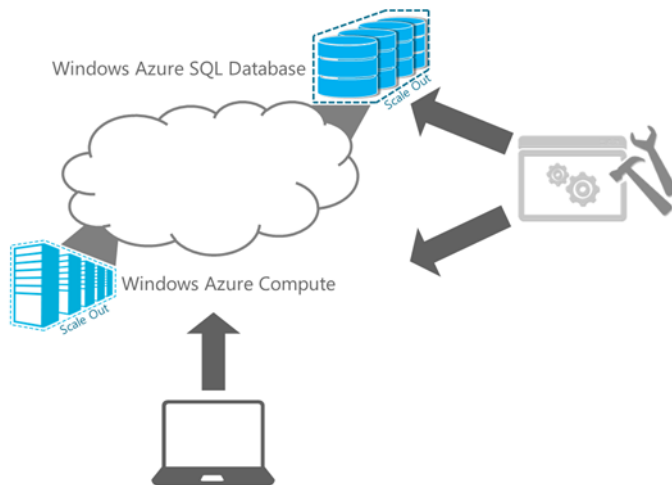
To take advantage of many of the unique capabilities of SQL Database, an application must be specifically architected and developed for those features. For example, to use SQL Federation applications, developers must understand the constraints presented by the sharding scale-out pattern and must know how to access shard members through USE FEDERATION T-SQL statements.

Developers typically can build a new application for the cloud in SQL Database by undertaking all development work while connected to a remote database running in Windows Azure SQL Database. Organizations typically create their servers and databases from the Windows Azure or SQL Database management portals. Then, they connect their existing tools to those remote databases. As noted earlier, SQL Database is highly compatible with SQL Server and as such supports most tools based on SQL Server, including Visual Studio and SQL Server Management Studio 2012. Developers use these tools, working against the remote server, to build out their database schema.

Usually, developers build the application tier by using Windows Azure Cloud Services or Windows Azure Web Sites features. With these tools, application developers can build applications that can be more easily scaled out behind Windows Azure load-balancers. Scale-out in the application tier closely matches the approach of scaling out the data tier, and this means that developers can build applications that meet the demands of massive workloads in the cloud. Developers are not restricted to just the Microsoft .NET framework; Cloud Services work well with Java, and SQL Database works well with the JDBC drivers for SQL Server. Microsoft also supports an open-source project that provides Windows Azure add-in tooling for Eclipse, a popular Java IDE. Other frameworks such as Ruby, PHP, and Node.js are equally well supported by SQL Database and Windows Azure.

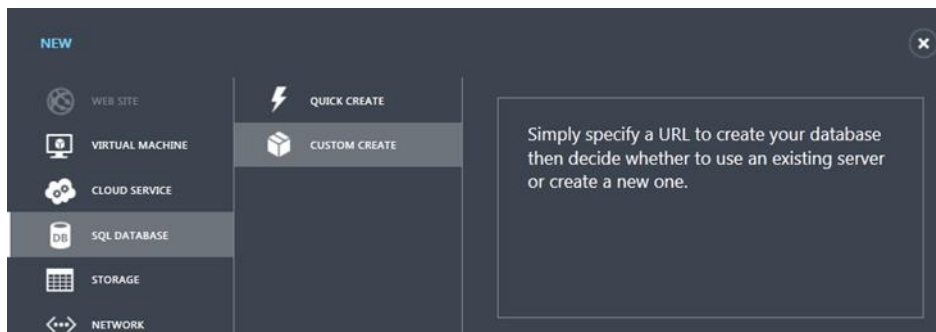
IT pros use a combination of Windows Azure and SQL Database portals and existing SQL Server management tooling to manage the ongoing operation of the database (Figure 26).

Figure 26: In SQL Database, developers can scale out applications in the app and data tiers



You can use the Windows Azure Management Portal to create a SQL Database. Use the Custom Create option to create a SQL Database server (Figure 27).

Figure 27: Using the Custom Create option for a new SQL Database



Note that a *server* in SQL Database is a logical concept only; the actual database replicas that make up any given SQL Database instance are physically located in a number of different physical nodes within a data center (Figure 28).

Figure 28: Specifying a new server when creating a database

The screenshot shows the 'Specify database settings' form in the Windows Azure Management Portal. The form is titled 'NEW SQL DATABASE - CUSTOM CREATE' and 'Specify database settings'. It contains the following fields and options:

- NAME:** A text input field containing 'WhitePaperDemoDB'.
- EDITION:** Two radio buttons, 'WEB' (selected) and 'BUSINESS'.
- MAXIMUM SIZE:** A dropdown menu with '1GB' selected.
- COLLATION:** A dropdown menu with 'SQL_Latin1_General_CP1_CI_AS' selected.
- SUBSCRIPTION:** A dropdown menu with 'C' selected.
- SERVER:** A dropdown menu with 'New SQL Database Server' selected.

Figure 29: Creating a new server, including log-in details

NEW SQL DATABASE - CUSTOM CREATE

Database server settings

LOGIN NAME

admin_user

LOGIN PASSWORD

LOGIN PASSWORD CONFIRMATION

REGION

East Asia

☒ Allow Windows Azure services to access the server

SQL Database includes a firewall that protects against attempted connections from unknown IP addresses. You must create firewall rules (Figure 30) to explicitly open access for each address or address range that is expected to connect to the database server (Figure 31). Select **Windows Azure services** to set firewall rules that permit other services in Windows Azure, such as Windows Azure Compute and Windows Azure Virtual Machines, to access the database server. To connect by using Management Studio or other tools from on-premises machines, you must create additional rules.

Figure 30: Adding firewall rules (click the server hyperlink to open the server properties)

sql databases

DATABASES SERVERS

NAME	STATUS	LOCATION	SUBSCRIPTION	SERVER	EDITION	SIZE
WhitePaperID	Online	East Asia	Microsoft Azure	Database325	Web	1 GB

Figure 31: Adding firewall rules (specifying a range of permitted addresses)

allowed ip addresses

My Work Desktop 203.97. 203.97.

RULE NAME START IP ADDRESS END IP ADDRESS

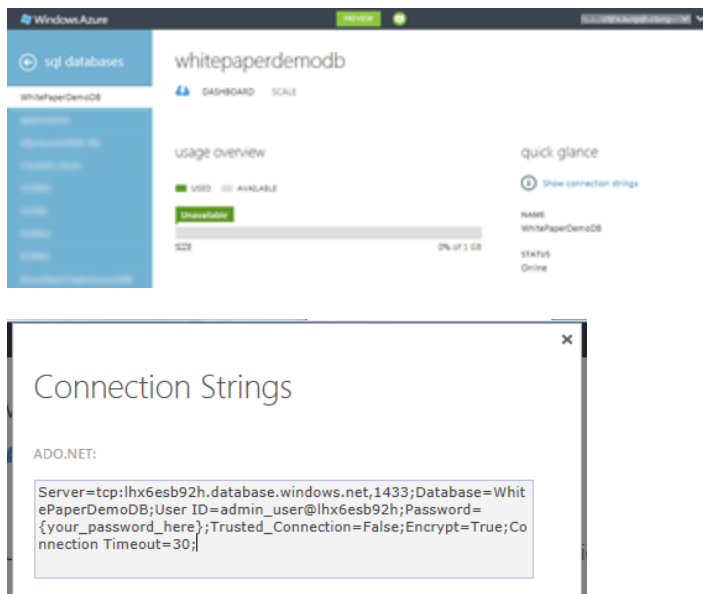
CURRENT CLIENT IP ADDRESS 172.20.0.1

allowed services

WINDOWS AZURE SERVICES YES NO

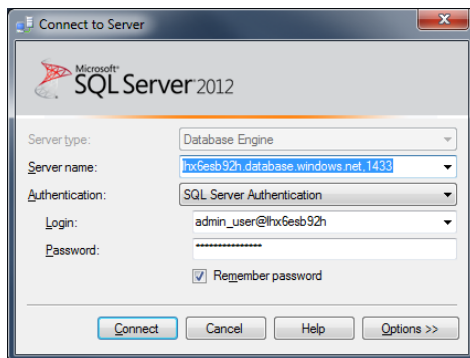
After the server is configured, you can retrieve the connection strings from the database properties within Management Portal (Figure 32).

Figure 32: Using the connection string to connect Management Studio (getting information)



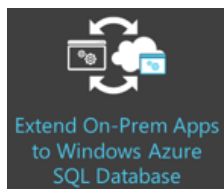
Use the information contained in the connection string to connect SQL Server Management Studio to the remote database.

Figure 33: Using the connection string to connect Management Studio (entering settings)



From here, use the familiar tooling available in Management Studio to create database schema. SQL Server Management Studio also supports creation and management of Federations.

Extend On-Prem Applications to Windows Azure SQL Database



In some situations, organizations may want to move just a portion of an application to the cloud and retain sensitive data on-premises while moving portions of the application that require additional scale to the public cloud.

Consider the e-commerce site. Sensitive information such as customer details and credit card information may need to remain on-premises; but, to support increased load, the product catalog and shopping cart data could be moved into Windows Azure.

We discuss these types of hybrid applications in more detail in next sections.

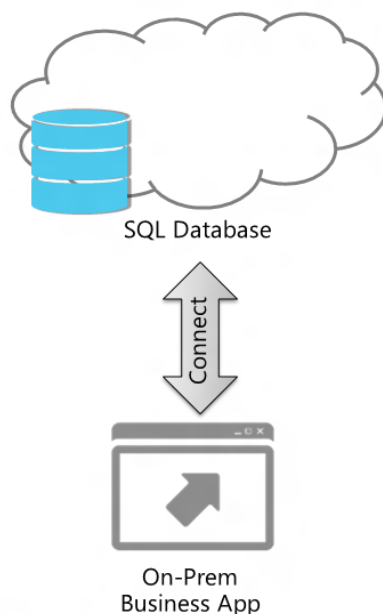
Hybrid scenarios

In addition to giving organizations choices about where they run their apps, the concept of hybrid IT means they also can choose to distribute their apps across the public cloud and private data center. Windows Azure Virtual Network allows creation of a logically isolated section in Windows Azure and securely connect it to on-premises datacenter or a single client machine using an IPsec connection. Virtual Network makes it easy to extend datacenter using Windows Azure much in the same way of setting up a remote branch office. Organizations retain control over the network topology and configuration, and manage it in the same way as on-premises infrastructure. Virtual Network makes it easier to build hybrid scenarios while maintaining secure connections with on-premises infrastructure without the creation of custom codes.

Code-far apps: A simple path to an enterprise-class data tier

SQL Database offers many enterprise-class database features, including high availability. Many departmental applications would benefit from these capabilities, but often it is too expensive to deliver such features on-premises. A *code-far* hybrid approach involves connecting an on-premises application over the Internet to a SQL Database instance (Figure 34).

Figure 34: Providing an enterprise-class database for lightweight apps through a remotely accessed SQL Database

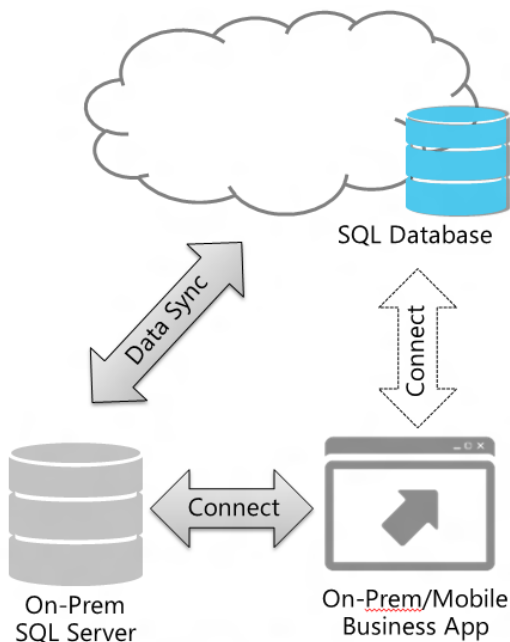


Microsoft Access 2013 presents an excellent example of this deployment architecture. Access 2013 supports developers in building Access 2013 apps that run either in Microsoft Office 365 or on-premises servers, but store their data in a Windows Azure SQL Database.

Apps occasionally connected with SQL Data Sync

SQL Data Sync is a service built on the Microsoft Sync Framework. Organizations can use it to synchronize data bidirectionally between on-premises SQL Server and SQL Database instances. Because SQL Data Sync is provided as a service within Windows Azure, there is no need for developers to write custom code; they can configure SQL Data Sync in the Azure portal and then install the SQL Data Sync Agent in the on-premises servers (Figure 35).

Figure 35: Supporting occasionally connected work through hybrid architecture by using SQL Database and SQL Data Sync

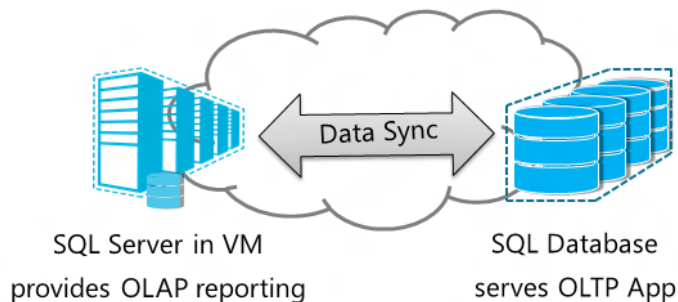


This hybrid architecture allows organizations to build applications that support occasionally connected work. When the user is not connected, data is stored in a local SQL Server database and then synchronized with the SQL Database hub once connectivity is restored.

SQL Database plus SQL Server in Windows Azure Virtual Machines—in a single app

As described earlier, SQL Database and SQL Server in a Windows Azure Virtual Machine have respective strengths; another hybrid approach is to combine both into a single application. As an example, an organization may want to take advantage of the scale-out capability of SQL Database for an OLTP web application. They also may need to perform multidimensional analysis and reporting, a capability that would require the full functionality of SQL Server installed into a virtual machine (Figure 36).

Figure 36: Enabling apps to use strengths of deployment options (SQL Database combined with SQL Server)

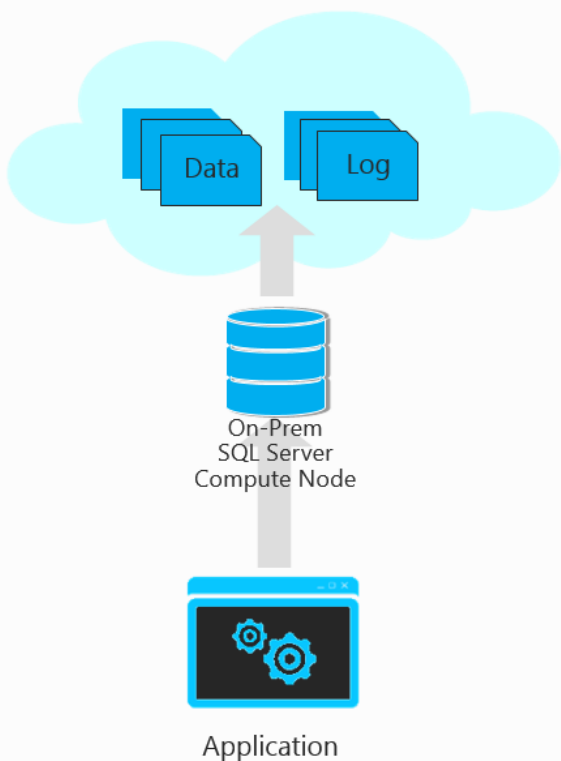


By using SQL Data Sync, organizations can deploy SQL Database to drive the transaction processing needs of the application and use the full power of SQL Server Reporting and Analysis Services on-premises to provide the analytical processing requirements.

SQL Server Data and Log Files in Windows Azure Storage

New in SQL Server 2014, organization can move data and log files from running database into Windows Azure Storage, while keeping the compute node of SQL Server running on-prem (Figure 37). This can be easily done without the need of changing the applications so no application compatibility testing is required. For added security to the data and log files, Transparent Data Encryption can be turned on for the database and the encryption key can be stored on-premise.

Figure 37: SQL Server Data and Log Files in Windows Azure Storage



By using this scenario, customers can keep their applications intact while enjoying virtually 'unlimited' storage capacity in Windows Azure Storage, built-in Windows Azure Storage high availability, built-in multiple copies on different geographical locations for added protection, and faster recovery in case of on-premise disaster.

Conclusion

Cloud computing provides new opportunities for organizations to deploy applications more cheaply or at greater scale than ever before. At the same time, Microsoft hybrid IT strategy recognizes that most organizations typically have a range of applications, some of which are deployed to the cloud and some of which remain on-premises. Complex applications that require detailed hardware configurations and optimization, or that house sensitive data, are not well suited to the types of commodity services delivered

by cloud computing; these applications are likely to remain on-premises for some time to come. At the other extreme, some workloads are well suited to public cloud deployment—applications with highly variable demand, for example. The goal of Microsoft hybrid IT is to offer choices for organizations to use the same industry-leading technology, techniques, and expertise across on-premises servers, private clouds, and public cloud platforms.

Hybrid IT delivers the power of SQL Server, an industry-leading DBMS, across the full spectrum of deployment topologies. It delivers the same familiar SQL Server experience and tool set, regardless of whether an organization chooses to deploy on-premises or into a public or private cloud.

With Microsoft SQL Server in a Windows Azure Virtual Machine, organizations can take advantage of the efficiencies of cloud computing while also experiencing almost full feature parity with on-premises SQL Server deployments. Organizations that want to use the cloud to deliver new applications to large-scale audiences can use Windows Azure SQL Database to build next-generation relational database applications that scale out to millions of users.

Whatever the specific needs of your application scenario, and wherever it may be deployed, your organization can be confident that Microsoft provides the capability, flexibility, and familiarity you need.

For more information:

<http://www.microsoft.com/sqlserver/>: SQL Server Web site

<http://technet.microsoft.com/en-us/sqlserver/>: SQL Server TechCenter

<http://msdn.microsoft.com/en-us/sqlserver/>: SQL Server DevCenter

Did this paper help you? Please give us your feedback. Tell us on a scale of 1 (poor) to 5 (excellent), how would you rate this paper and why have you given it this rating? For example:

- Are you rating it high due to having good examples, excellent screen shots, clear writing, or another reason?
- Are you rating it low due to poor examples, fuzzy screen shots, or unclear writing?

This feedback will help us improve the quality of white papers we release.

If you have specific questions in these areas, or any of the areas discussed in the current paper, that you would like to see addressed in the subsequent paper, please submit them through the feedback link.

[Send feedback.](#)