Overview of Microsoft Azure

Author:	Holger Sirtl
Version:	2.7
Azure SDK:	2.7.1
Last updated:	September 1, 2015

© 2016 Microsoft Corporation. All rights reserved. Microsoft, Windows, Windows Vista and other product names are or may be registered trademarks and/or trademarks in the U.S. and/or other countries.

The information herein is for informational purposes only and represents the current view of Microsoft Corporation as of the date of this presentation. 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 provided after the date of this presentation.

MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS PRESENTATION.

Information in this document, including URL and other Internet Web site references, is subject to change without notice. This document supports a preliminary release of software that may be changed substantially prior to final commercial release, and is the proprietary information of Microsoft Corporation.

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

The entire risk of the use or the results from the use of this document remains with the user. 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.

Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, email address, logo, person, place or event is intended or should be inferred.

© 2016 Microsoft Corporation. All rights reserved.

Microsoft, Internet Explorer, Windows, and Visual Studio are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

Table of Contents

© 2016, Microsoft Deutschland GmbH

List of Figures

1 Cloud Computing with Microsoft Azure

Microsoft Azure is Microsoft's public cloud platform for developers and administrators who want to execute their own software applications partially or entirely in the cloud. The platform includes a range of cloud services that can be used individually or in combination and are provided on-demand in nearly any capacity and are charged according to usage. These services include, inter alia, those that are used to execute applications in the cloud, others that are used to store and efficiently process various data in the cloud, and yet others that can be used to securely and effectively link cloud elements with resources from a customer's data center or locally executed client applications.

A major strength of Microsoft Azure actually rests in such hybrid solutions in which individual application components are operated together with other components in a local execution environment. For example, a conventional Windows rich client application could store data in Microsoft Azure Storage, apps on mobile devices could exchange messages among each other via Microsoft Azure, a locally operated SQL server database could synchronize partially or fully with a cloud database, and a Web application operated in Microsoft Azure could access a locally executed backend system via the Service Bus. There are many other conceivable hybrid scenarios. Applications can, of course, also be operated entirely on Microsoft Azure.

Various development and management environments (Visual Studio, Eclipse, System Center, etc.) and programming technologies (.NET, Java, PHP, Python, etc.) with corresponding APIs can be used to create and migrate Azure-based applications. This allows developers to draw on their existing knowledge and expertise.

1.1 Cloud computing

The term "cloud computing" is often defined normatively, i.e., according to certain properties that are generally associated with cloud computing. Microsoft Azure fulfills these properties and can therefore be rightly referred to as a cloud platform:

- Azure services can be provided in nearly any capacity without requiring any upfront investments. Capacity planning at the start of a software project is greatly simplified: the application can be started in a small, affordable environment and then additional resources can be added quickly according to load.

- The services are provided on-demand, i.e., they can be used immediately once there is a need for compute, storage, or communication services, and the services can be deactivated once there is no more demand. This is a critical advantage over a local infrastructure when, for example, using the cloud to set up test and development environments.
- Billing is usage-based, i.e., costs are only incurred for the capacity actually utilized. This means that the costs are correlated with the success of a cloud-based software system, and the risk of expensive bad investments is eliminated.
- Standard interfaces (e.g., RESTful services) allow these services to be used across all technologies supporting these standards (e.g., .NET, Java, PHP).

1.2 Structure of the Microsoft Azure platform

Fig. 1-1 shows the rough structure of the Microsoft Azure platform. The platform consists of numerous highly scalable, quickly provisionable, flexibly administrable services provided on Microsoft's cloud infrastructure. These services can be addressed and administered via a Management API (REST-based), i.e., custom environments consisting of these services or instances of services can be created, configured, scaled, and deleted.

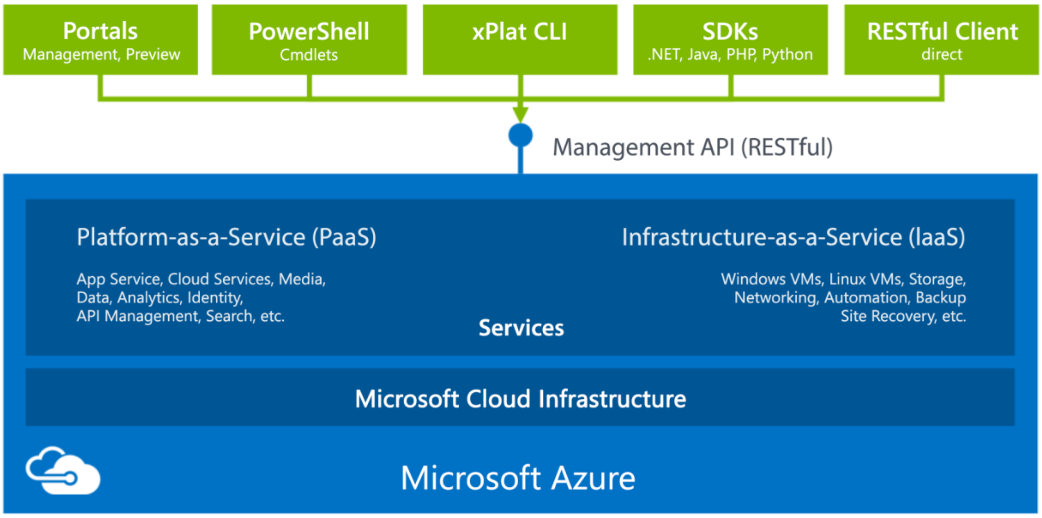

Portals
Management, Preview
PowerShell
Cmdlets
xPlat CLI
SDKs
.NET, Java, PHP, Python
RESTful Client
direct
Management API (RESTful)
Platform-as-a-Service (PaaS)
App Service, Cloud Services, Media, Data, Analytics, Identity, API Management, Search, etc.
Infrastructure-as-a-Service (IaaS)
Windows VMs, Linux VMs, Storage, Networking, Automation, Backup Site Recovery, etc.
Services
Microsoft Cloud Infrastructure
Microsoft Azure

Fig. 1-1 Rough structure of the Microsoft platform

Because the Management API is implemented according to the REST paradigm, it can be addressed by any technology supporting calls of RESTful services. It can therefore be used without problem from .NET, PHP, Java, Python, etc. More convenient access to the API is available via Software Development Kits (SDKs) for various programming languages. These enclose REST calls in objects of the respective language, enabling developers in the supported languages to use classes in their familiar implementation technology to control Azure Services. For those who prefer using the command line, there are cmdlets for the Windows PowerShell and, for Windows, Linux, and Mac environments, the cross-platform Command Line Interface (xPlat CLI). Azure's REST interfaces can be addressed from the command line using these tools.

Beginners, however, will probably gravitate first to one of the two Azure Web portals: the Management Portal and the Preview Portal. The Preview Portal will completely replace the Management Portal in the mid-term future. The portals expose large parts of Azure's functionality via a Web-based interface. They are also able to address the REST API.

1.3 Microsoft's cloud infrastructure

Microsoft's cloud infrastructure forms the basis for all other services. It consists of server and network components that ensure reliable operation of the platform, are retained in Microsoft's data centers, and are quickly deployed when needed. Using the Microsoft Azure Pack, individual Azure services (e.g., Web Apps and Virtual Machines) can be operated on a Windows Server environment in a customer's data center.

1.4 Services of the Microsoft Azure platform

Various services in the categories of Infrastructure-as-a-Service (IaaS) and Platform-as-a-Service (PaaS) are available for execution of custom application logic. The user decides which operational aspects are handled automatically by the platform and which are managed by the user itself. At the IaaS level, pre-built or custom Windows- or Linux-based virtual machines can be run on Azure. PaaS includes cloud services for more complex, multilayered Web applications, Web Apps for fast, flexible Web hosting, mobile apps as a backend for mobile applications, and media services for implementation of media workflows.

Further services can be used from custom applications (whether executed on Azure or locally). The Identity area provides services for authentication and user administration. Integration services can be used for communication and integration of distributed application components, and there is a full range of different storage services available for data storage and evaluation. Annex A lists the services of the individual functional groups and refers to the respective sections in this document.

The services provided by Azure are designed for high availability. They are constantly monitored and kept up to date. Failures are detected, and appropriate resolution measures are initiated automatically. The services can respond to a fluctuating load, i.e., on request an auto-scaling mechanism can add or remove resources automatically. Management tasks that users want to execute can be controlled via a Web portal, PowerShell scripts, or REST interfaces.

Relevant developer tools are provided for all the provisioned services and greatly simplify the application lifecycle, starting from creation to testing and through to deployment and operation. Various other platforms and tools (Node.js, Java, PHP, Python, Ruby, Eclipse) with corresponding extensions are supported in addition to the Microsoft development platforms (.NET and Visual Studio). Support is also provided for interaction with various source code management systems (Visual Studio Online, GitHub, etc.).

1.5 Structure of this document

The individual chapters of this document address the following topics:

- Chapter 2: Providing Custom Application Logic on Microsoft Azure
  This chapter shows how custom applications and services can be executed and managed on Microsoft Azure. Several execution models are available for this purpose. These are, inter alia: App Service, Cloud Services, and Virtual Machines.

- Chapter 3: Storing Data on Microsoft Azure
  Microsoft Azure offers various relational and non-relational storage services. This chapter provides an overview and shows when it makes sense to use which service.

- Chapter 4: Analytics and Big Data Services
  Azure services suitable for storage, evaluation, and research of even the largest datasets are presented in this chapter.

- Chapter 5: Security of Azure-based Applications
  This chapter presents services and service descriptions that help make custom applications secure and document their security.

- Chapter 6: Media Processing and Distribution
  The services presented in this chapter are focused on media processing (audio, video, images, etc.) and media distribution.

- Chapter 7: Integration of Azure-based Applications into In-house IT
  This chapter is devoted to bridging the gap between Azure and a customer's data center. It presents services that can be used to implement hybrid cloud environments.

- Chapter 8: Network Services
  Various services for networking virtual machines on Azure among one another, their connections from the Internet, or to one or more local data centers are presented here.

- Chapter 9: DevOps with Microsoft Azure
  This chapter focuses on services for development and operation of cloud-based applications and their integration into an integrated DevOps solution.

- Chapter 10: Getting Started with Microsoft Azure
  References to information sources, training, videos, etc. that may be helpful in familiarization with Microsoft Azure are provided here.

2 Providing Custom Application Logic on Microsoft Azure

Those who approach the issue of cloud computing with Microsoft Azure often think first of the ability to execute applications in the cloud, i.e., Microsoft's data centers. While this is not the sole function of Microsoft Azure, it is a central one. Further services of the platform, such as storage, database, or Service Bus – even if they could be used alone – can then be smoothly combined with the applications executed on Microsoft Azure.

Microsoft Azure provides three models for executing applications in the cloud; these are outlined in Fig. 2-1: Web Apps, Cloud Services, and Virtual Machines. These can be used independently of one another or in combination. There are also migration paths from Web Apps to Cloud Services and from Cloud Services to Virtual Machines. Web Apps offer an affordable hosting option for Web applications; Cloud Services are ideally suited for setting up and operating highly scalable, highly available, multilayered applications with minimum administrative overhead; and Virtual Machines allow operation of custom or pre-built virtual machines in the cloud.

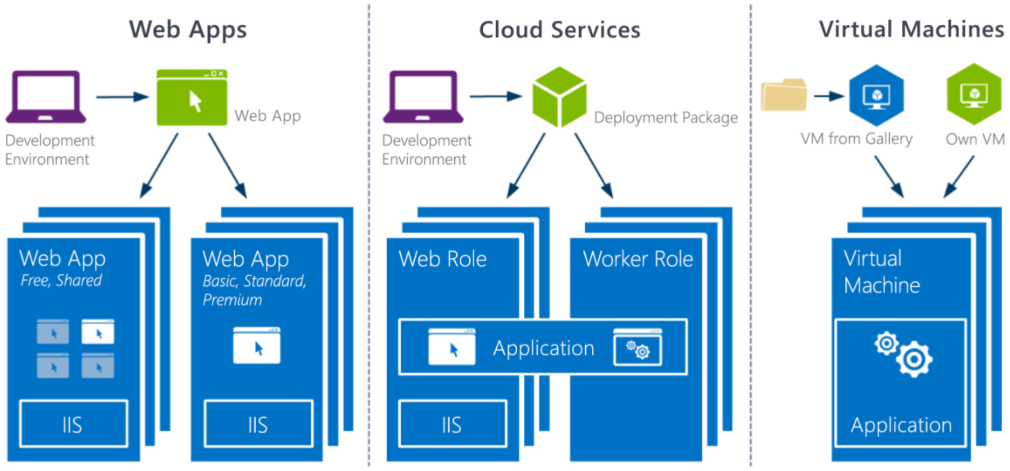


Fig. 2-1 Execution models for cloud applications

Virtual machines are ultimately used in all three alternatives, with differences in the extent to which administrative tasks are handled automatically by Azure. Virtual Machines offer the greatest flexibility in configuring virtual machines, leaving many administrative tasks (patching, upgrades, etc.) to the user of Virtual Machines. Web Apps, on the other hand, handle very many tasks for the user (guest operating systems are maintained automatically, required runtimes and frameworks can be installed at the push of a button) and the functionality is more limited to a defined standard. The following table compares the execution models.

	Web Apps	**Cloud Services**	**Virtual Machines**
Level (PaaS/IaaS)	PaaS with the portability of IaaS	PaaS	IaaS
Managed artifact	Web App (e.g., ASP.NET Web site)	Cloud Service	Virtual machine
Portability	Fully portable	Design or adapt for the cloud	Fully portable
Management	Automated	Automated	Manual
Architecture	2-layer IIS Web sites	Flexible	Flexible
Persistence	Persistence at VM level	No VM persistence (persistent drives supported)	Each VM is persistent
Provisioning	< 20 seconds	8-10 minutes	8-10 minutes
System platform	Windows Server / IIS	Windows Server	Windows Server, Linux
Deployment	Web Deploy, FTP, Git, Visual Studio Online	Packaging and publish	Creation and upload of VHDs or selection from a gallery
Gallery	Web frameworks (e.g., WordPress, PHP)	Microsoft OS images (Web Roles and Worker Roles)	Microsoft or custom OS images

Table 1: Comparison of execution models under Microsoft Azure

2.1 Hosting Web-based application logic with the App Service

The App Service provides functions for a group of application types that have similar requirements in terms of installation, provisioning, authentication, scaling, etc. In addition to Web Apps for hosting conventional Web applications, these include Mobile Apps for providing backends for mobile clients, Logic Apps for implementing API-based workflows, and API Apps providing functions over Web-based interfaces.

2.1.1 Hosting Web applications with Web Apps

Microsoft Azure Web Apps are (in addition to Cloud Services and Virtual Machines) one of three options for executing Web applications in the cloud. They provide an affordable but well scalable hosting option for Web sites. Through the Microsoft Azure Portal, environments can be set up very quickly and easily, preconfigured with numerous Web frameworks (e.g., ASP.NET, Java, PHP, Node.js) and applications (e.g., Joomla, WordPress, Drupal), and loaded with Web applications from various development platforms. Deployment options such as Git, FTP, WebDeploy, and TFS[1] Deploy are supported. The portal offers various deployment scripts that can be imported into custom development environments. This allows for Web sites to be developed in Windows environments as well as non-Microsoft environments such as Linux or MacOS and then operated on Microsoft Azure. Support is also provided for automated deployment directly from source code management systems (including Git, Visual Studio Online, Dropbox, SkyDrive, etc.). A new deployment (that can be reverted on the portal) is performed automatically once a developer sets new code, so that a Web site always uses the latest source code.

Several operating modes are available for executing Web Apps: in the "Shared" model, one or more instances of the application are operated in a multi-user environment, the "Free" model is limited to a single instance, and operation in the "Basic" model is exclusively in virtual machines provided for the application. The "Standard" and "Premium" models additionally offer Backup and Restore, Auto-Scaling, WebJobs, Scheduler support, and Staged Publishing through a staging environment.

The operating modes can be switched flexibly, allowing both scale-out (activation of additional instances of the same size) and scale-up (increase of the existing instances in the "Basic" or "Standard" models) to adjust the environment to changing loads. This can also be automated in the "Standard" and "Premium" models using the AutoScale function (e.g., according to CPU load). The configuration and management of the virtual machines used to execute Web Apps are handled completely by Microsoft Azure.

MySQL and Microsoft Azure SQL Database are available as databases for persisting data. The other services of the Microsoft Azure platform can of course also be used from Web Apps. For example, a Web App can authenticate users via the Azure Active Directory or access an SQL server executed in a virtual machine to store data.

➔ http://azure.microsoft.com/en-ie/services/app-service/web/

[1] Team Foundation Server (component of the Visual Studio platform)

2.1.2 Backends for mobile clients with Mobile Apps

"Connected mobile Apps," i.e., lightweight applications executed on mobile devices, are often networked via backend services. The individual clients exchange data, store data, perform authentication, etc. via these services. Microsoft Azure Mobile Apps offer a collection of exactly these frequently needed services on Microsoft Azure. These services include, inter alia, the following functions:

- User authentication (via Microsoft Account, Facebook, Twitter, and Google ID) with appropriate access control for service calls
- Storage of data in Blob, Table, Queue Service, or SQL Database
- Communication via Service Bus
- Push notifications
- Scheduled execution of server-side scripts

It is also possible to provide custom interfaces in addition to data-based interfaces (for writing, reading, modifying, deleting data).

The preconfigured Microsoft Azure Mobile Apps can be deployed very easily, without major programming effort, as backend services for Windows 8.x, Windows Phone 8.x, Windows 10, iOS, Android, and Web clients. Templates for relevant new Apps or code fragments for existing Apps can be downloaded from the portal. This greatly simplifies the development of simple mobile applications that use cloud services.

http://azure.microsoft.com/en-ie/services/app-service/mobile/

2.1.3 API-based workflows with Logic Apps

Using Logic Apps, workflows can be defined that consist of a sequence of API calls and the corresponding progressive processing of data, i.e., returns from an API can be used as call parameters for subsequent APIs. This enables the implementation of business processes that can include custom application logic (using independently developed APIs) and logic from various backend systems (via pre-built connectors loaded from the Marketplace and appropriately configured).

http://azure.microsoft.com/en-ie/services/app-service/logic/

2.1.4 Web-based interfaces with API Apps

Application logic often needs to be provided flexibly in the form of a callable API. Although it is quite easy to create an API using technologies such as Web API, functions for authentication, calling, etc. often require effort to implement. API Apps make it much easier to provide these accompanying functions. When implementing clients that are to use an API App, metainformation (call parameters, return values, authentication requirements, etc.) can be imported using a Swagger-based description and the code for the calls can be automatically generated. This lets developers focus completely on the application logic and use objects from their respective programming language.

Developers can either implement API Apps entirely on their own or provision them by configuring appropriate connectors. Azure already contains numerous connectors in the Marketplace that allow, for example, sending e-mails, setting data in Salesforce, or making tweets on Twitter.

➔ http://azure.microsoft.com/en-ie/services/app-service/api/

2.2 Multilayered applications with Cloud Services

Microsoft Azure Cloud Services are (in addition to Web Apps and Virtual Machines) one of three options for executing applications in the cloud. Cloud Services focus very strongly on highly available, highly scalable, multilayered Web applications that benefit most from the flexible resource provision of a public cloud. These applications can be divided into components, called "roles," that can be scaled independently of one another. Microsoft Azure handles the management of the virtual machines required to execute the roles. This includes upgrades to guest operating systems, patching, and failover (failed components are automatically replaced). Unlike with Web sites, developers have full administrative access to the virtual machines in order to, for example, install additional software in the individual instances. Remote Desktop connections for administration are also supported. Roles can be scaled independently of one another (addition or removal of instances); this operation can also be automated (depending on various load parameters) using the auto-scaling function.

Development for Cloud Services can be done in Visual Studio or Eclipse. The applications can be executed locally in an Azure emulator and, following successful testing, deployed to the cloud directly from the development environment. Cloud Services thus offer application development and operation under the PaaS (Platform-as-a-Service) model. Cloud Services provide greater control of deployment and distribution of application components than do Web Apps.

Microsoft Azure-based Cloud Services are ultimately executed in the cloud on Windows Server-based virtual machines (VMs). How Microsoft Azure configures these VMs depends on the structure of the cloud service, the selected instance sizes, and the selected VM versions.

The developer can define the structure using a cloud service definition (csdef) and a cloud service configuration (cscfg). The deployment package, which can be created by Visual Studio, Eclipse, or command-line tools, contains all the components required to execute the application (application code, libraries, frameworks, server systems, etc.). The rest is handled by Microsoft Azure. If the number of instances changes at runtime (due to change in configuration or to usage of the AutoScale function), instances are automatically shut down or additional instances are provisioned, respectively, and the upstream load balancer is reconfigured accordingly. The administrative effort for the user is thus kept to a minimum. The costs for operation are billed per hour. If new virtual machines are created (e.g., for a large patch or in case of hardware failure), their state is reset to that at the time of deployment, i.e., the virtual machine is created from a base image and the deployment package. Unlike with Virtual Machines, the virtual machines in Cloud Services are non-persistent. Applications executed in Cloud Services should thus save their state outside of the virtual machines (e.g., in Microsoft Azure Storage or SQL Database) and use the drives within the virtual machines only for caching.

http://azure.microsoft.com/en-ie/services/cloud-services/

2.3 Microservices-based Apps with Service Fabric

The Service Fabric is the latest member in the Compute Services family. Currently it has only been announced. This service will be the fourth option for implementing cloud-based applications in Azure (in addition to App Service, Cloud Services, and Virtual Machines). Apps written for the Service Fabric differ from Apps in other compute modes in terms of structure and deployment. With Service Fabric, applications can be made up of microservices that can be installed dynamically and distributed to a pool of provisioned virtual machines. The instances of a microservice can share the resources with other microservices, i.e., more than one microservice instance can be executed on a virtual machine. An advantage over the other execution options is the significantly more flexible resource allocation and better utilization of the machines. Applications can scale more flexibly, specifically at the microservice level. Individual functions can be packaged into microservices and then scaled very individually (if, e.g., a load peak occurs only in a certain application function). However, applications have to be written very specifically for the Azure Service Fabric.

http://azure.microsoft.com/en-ie/documentation/services/service-fabric/

2.4 Operating virtual machines using Virtual Machines

Virtual Machines introduce the IaaS element to Microsoft Azure. They allow the operation of virtual machines on Microsoft Azure that were created by the user or provided by Microsoft. These are charged by the minute. Provisioning a virtual machine requires only the specification of a VHD (standard VHDs are used here) for the booting process, the VM size, and a few configuration parameters (e.g., administrator account, DNS name) depending on the selected VHD.

Hard disk drives used in the virtual machines are persistent in storage as page blobs. A virtual machine should be reimaged so that all content is retained (e.g., in case of hardware failure). The same failover rules apply to the content (storage in triplicate of all content, optional georeplication to a remote data center) as for all other content in Blob storage. Because Microsoft Azure uses standard VHDs, existing VMs can be migrated to the cloud from a local Hyper-V environment and executed as virtual machines, and conversely VHDs can be downloaded from Microsoft Azure and executed locally in an appropriate Hyper-V environment.

Various Windows Server versions and selected Linux distributions are supported as guest operating systems for virtual machines. Several server applications such as SQL Server, SharePoint Server, BizTalk Server, and Active Directory are likewise supported. Due to cooperation with Oracle, applications from Oracle's technology portfolio (WebLogic, Java, Oracle DB, etc.) are supported. It is possible to create new VHDs in a local Hyper-V environment, upload them to Microsoft Azure, and boot new virtual machines from the VHDs. Pre-built images with corresponding server software from a VM gallery can also be selected and started. MS Open Tech maintains various VM images with open source stacks in the VM Depot that can then be easily imported into Azure and executed there.

This makes it possible to migrate important parts of the server environments for a customer's data center to Microsoft Azure without modification. The minute-based billing also makes this execution model interesting for test or demonstration environments where infrastructure is often only temporarily required.

Like the other execution models, Virtual Machines can also be used in combination with other Azure services. For example, an application executed in a VM can smoothly use Microsoft Azure Storage or SQL Database or handle user authentication via the Azure Active Directory. The combination with a virtual network (see 8.1) is particularly interesting: The ability to set up a virtual network spanning local and cloud VMs with a shared IPv4 address space and corresponding routing tables makes it possible to move virtual machines back and forth flexibly between the cloud and a local data center without affecting the other VMs from a network perspective.

➔ http://azure.microsoft.com/en-ie/services/virtual-machines/

2.5 Batch processing with Azure Batch

The automated execution of large computational processes (e.g., for mathematical calculations, video rendering, data analyses) normally poses several challenges: the temporary provision of sufficient compute capacity and distribution of the computation logic and associated data are difficult or impossible to implement in conventional IT. Azure Batch can handle these tasks, making the capabilities of batch processing available to every Azure user. The service provides the cluster environments required, distributes the calculation logic provided by EXE file or script, manages the execution, and releases the computational resources following completion so that only the required resources are billed.

➔ http://azure.microsoft.com/en-ie/services/batch/

2.6 Managing mobile Apps with Mobile Engagement

Mobile applications can be implemented quickly and efficiently with Mobile Apps and the templates for clients on various platforms. While these applications are operational, the user may often be interested in which of the applications' functions are frequently used, where problems occur, etc. Mobile Engagement allows access to such information via a dashboard. It can also be used to send messages to users to, for example, share campaign-related information.

➔ http://azure.microsoft.com/en-ie/services/mobile-engagement/

2.7 Push notifications with Notification Hubs

The ability to send push notifications to a large number of different clients specifically, quickly, and easily is a major challenge in modern, mobile application scenarios. Notification Hubs make it possible to send messages to clients on various platforms (Windows Store, Windows Phone 8, iOS, Android) with a single API call. The addressees can be set as individual users (who may have several devices), groups of users, or all users of associated client Apps.

This lets news portals, for example, send targeted current news matching their users' interests, or event sites can send schedule updates.

http://azure.microsoft.com/en-ie/services/notification-hubs/

2.8 Provisioning Windows Apps with RemoteApp

Client applications can be provisioned centrally via Azure using RemoteApp. These applications can then be called and executed from various client systems (Windows, Mac OS X, iOS, Android) via the Microsoft Remote Desktop protocol without storing the applications on the clients. This makes it possible to provision applications from a secure environment while also supporting a heterogeneous client system landscape.

http://azure.microsoft.com/en-ie/services/remoteapp/

2.9 API Management

Application logic is often provided as an API instead of via an interface usable by end users. API Management can help with marketing an API. Provisioned APIs can be assigned permissions, supplemented by converting input and output values, and supplied with product-specific SLAs. The usage of APIs can be monitored, evaluated, and billed accordingly as necessary. The provisioned API can be operated on Azure or externally.

http://azure.microsoft.com/en-ie/services/api-management/

3 Storing Data on Microsoft Azure

Secure data storage is a central requirement in almost every application. One option for data storage was already presented in Section 2.3: Storing in Virtual Machines. This can be done on the virtual machine's drives, on a preconfigured SQL server, or in any other storage system (including non-SQL systems such as MongoDB, Cassandra, etc.) operated in a virtual machine. Persisting the hard disks in Blob storage ensures that the data is securely stored.

Microsoft Azure also provides various storage types as real cloud services. Most aspects of managing these storage systems are handled by Microsoft Azure. There is SQL Database for relational data storage; DocumentDB and Table storage for non-SQL storage of semi-structured data; Blob Storage for large binary data (documents, videos, music, static Web site content); and Azure Disks as a persistent file system. The following table provides an overview of the various storage options.

	RDBMS in a VM	**SQL Database**	**DocDB**	**Table Storage**	**Blob storage**	**Files**	**Disks**
Info. type	Tables with schema		Objects	Key value	Blobs	Files	
Redundancy	Yes, 3x						
VM binding	Yes	No					
Optimization using indices	Multiple indices supported		Index per field	Only one index	N/A		
Interface	TDS		RESTful			RESTful, SMB	NTFS
Tx costs	Yes	No		Yes			
Use cases	Relational data storage and processing (SQL)		Semi-struct. JSON objects	Semi-struct. data with a key	Storing binary data (images, videos, etc.)	Frameworks requiring a persistent file system	

Table 2: Comparison of storage options under Microsoft Azure

All of these storage types (except for a VM's local file system) store data on distributed storage media in triplicate. Access is supported not only via applications operated on Microsoft Azure but also from outside, e.g., from a locally executed rich client PC application or a smartphone App. Content can be access-protected. Time-based access restriction is also supported.

3.1 Relational data storage

3.1.1 Relational database systems in virtual machines

An SQL server (or any other database system) operated in a virtual machine is an obvious option for performing relational data storage and processing in Microsoft Azure. The server configuration and administration are still handled by the user, who remains responsible for managing and configuring the server, installing updates, etc. and thus retains maximum influence on the server operation. Compatibility with a locally executed system is also provided, though the overhead for operation is comparable with operation of a local system.

3.1.2 RDBMS-as-a-Service: SQL Database

SQL Database integrates relational database technology as a true cloud service into the Microsoft Azure platform. This includes a relational database and a synchronization service that can be used to synchronize the content of multiple databases. The administrative overhead is kept to a minimum because it is mostly handled by Microsoft Azure.

With SQL Database, Microsoft Azure is equipped with a true relational database that is interface-compatible with SQL Server. Thanks to this compatibility, SQL Database can be accessed via familiar technologies (e.g., Entity Framework, ADO.NET, JDBC) and tools (e.g., SQL Server Management Studio). In many cases, addressing the target database using the connection string for a corresponding cloud database is sufficient.

Functionally, SQL Database maps a subset of SQL Server. The familiar database constructs such as tables, views, joins, atomic transactions, data integrity for competing accesses, etc. are available. The SQL security architecture is likewise supported, with the ability to create users and to assign them permissions.

In contrast with SQL Server (including an SQL server operated in a virtual machine), SQL Database is a true cloud service. Management of the environment (configuration of hardware and system software, monitoring, patch installation, etc.) is handled completely by Microsoft Azure. Setting up a new database server with an associated database takes a matter of seconds, and a database can be deleted just as quickly. Billing is based on the duration of provisioning and the size of the database.

SQL Database is offered in three editions: Basic, Standard, and Premium. Basic is well suited as an affordable alternative for low loads and database sizes up to 2 GB, Standard offers higher transaction performance for databases up to 250 GB in size, and Premium ensures maximum performance and transaction volume for databases up to 500 GB. One can switch between the editions easily at runtime in order to respond to changes in the load requirements.

Scalability beyond the limits of a single SQL database instance can be achieved with elastic scale for Azure SQL Database, a service that significantly simplifies scalability of the data layer in cloud applications. This is ultimately a sharding mechanism through which applications can access a pool of databases containing the distributed data set. Using a "contract," developers can define strategies specifying which data is ultimately stored in which database. This lets the service automatically determine which database is to be addressed for a specific transaction. Queries across multiple databases are also supported.

➔ http://azure.microsoft.com/en-ie/services/sql-database/

3.2 NoSQL data storage

In addition to relational, SQL-based data storage, Microsoft Azure also offers data services enabling NoSQL-based storage: DocumentDB as a document-based database with native support for JSON and JavaScript, Table Storage for storage of semi-structured data in tabular entity sets, Queue Storage[2] for sending messages between loosely, i.e., asynchronously, coupled communication partners, and Blob storage for storing and processing large binary data (BLOB = binary large object). The data is stored in a high-availability way in all cases (data is automatically stored in triplicate, with support for georeplication) and is available via RESTful interfaces. Microsoft Azure Disks are a special case: they provide an NTFS file system with corresponding file system interface on the basis of Blob storage. This allows for a persistent file system to be used in Microsoft Azure Virtual Machines.

[2] Queue Storage is addressed in the context of application integration in Section 7.2.1.

3.2.1 JSON object database DocumentDB

DocumentDB is positioned between SQL Database and Table Storage in terms of storing data. Formally, a non-schema NoSQL database allows DocumentDB to store more complex JSON-based objects. These are automatically indexed by the database and can be accessed using a query syntax related to SQL without specifying a schema or a secondary index. Native support for JavaScript and JSON makes DocumentDB suitable for development of applications that are based on these technologies and need to store the corresponding objects.

➔ http://azure.microsoft.com/en-ie/services/documentdb/

3.2.2 Table Storage

Microsoft Azure Table Storage is another NoSQL database in the Microsoft Azure platform. Despite what the name suggests, it is not a table-based relational database system but instead a memory for storing large semi-structured data sets in the cloud that require only a primary key for access and do not require complex relational database operations. From a single key (consisting of two mandatory attributes: partition key and row key), it is possible to read entities that consist of freely definable, typed properties (a property consists of a name, a type (e.g., string, integer), and a value).

Using the primary key values, Microsoft Azure can consolidate entries into partitions and distribute these automatically to various storage media to optimize performance. The data is automatically stored in triplicate, in a high-availability way, and can be accessed via RESTful interfaces. Alternative means of access include .NET, Java, and PHP class libraries. Although Table Storage has less functionality than SQL Database in terms of data management and complex access operations, it does provide affordable storage of semi-structured data and enables efficient, high-performance access to this data. Data sets of up to 500 terabytes per storage account are supported.

➔ http://azure.microsoft.com/en-ie/documentation/articles/storage-dotnet-how-to-use-tables/

3.2.3 Blob storage

Microsoft Azure Blob Storage enables the secure, highly available storage of large unstructured binary data (BLOB = binary large object) in Microsoft Azure. A single blob can have a size of up to 1 terabyte. Content is automatically stored in triplicate.

Other available options are georeplication and the ability to cache blob content distributed worldwide via the content delivery network (see 6.2). Two types of blobs are available: block blobs and page blobs. The former are suitable for sequential data upload and download, while the latter provide random access to Blob sections.

Blobs are stored within containers. Access privileges can be assigned both at the container level and the Blob level. The relevant element can be declared as public (access possible without authentication) or private. Shared Access Signatures can be created to time-limit access to appropriately authorized users.

https://azure.microsoft.com/en-ie/documentation/articles/storage-dotnet-how-to-use-blobs/

3.3 File system-based storage services

3.3.1 SMB 2.1 fileshares with File service

Using the File service, file shares can be provisioned via the standard SMB 2.1 protocol. This allows Azure-based applications and execution instances to easily exchange data among each other. This significantly simplifies migration of existing applications to the cloud because with SMB 2.1, a protocol that is often used in local execution environments can continue to be used.

3.3.2 Virtual hard disks in Virtual Machines: Disks

With Microsoft Azure Disks, data from Microsoft Azure Virtual Machines can be stored to a persistent file system. A VHD (virtual hard disk) stored in a Microsoft Azure page blob is used as the storage medium. Data stored there is – as usual for page blobs – stored in triplicate and is thus stored failsafe. Microsoft Azure Disks can be attached to or detached from virtual machines at runtime.

3.4 Caching

Applications often need to repeatedly access data in remote sources (e.g., SQL Database, Microsoft Azure Storage). To improve performance, it may be useful to cache this data in closer, higher-performance memory after the first read in order to process later requests more quickly. Microsoft Azure provides four mechanisms to achieve this: Redis Cache and Managed Cache Service as centrally provisioned caching services; In-Role Cache as an option to utilize unused resources within the components or dedicated instances as cache; and the Content Delivery Network, which can cache the content from Blob storage distributed worldwide.

http://azure.microsoft.com/en-ie/services/cache/

3.4.1 Azure Redis Cache

Redis Cache is a centrally provisioned caching service managed by Microsoft that is built on the open-source-based Redis Cache Framework. It can be addressed and used from any execution model (Web Sites, Cloud Services, Virtual Machines) and it is recommended for all applications to be developed for that cloud that require a cache.

➔ https://azure.microsoft.com/en-ie/documentation/services/redis-cache/#redis

3.4.2 Managed Cache Service

The Managed Cache Service provides a centrally usable in-memory cache. In contrast with In-Role Cache, which is reserved for Cloud Services, the Cache Service can (like the Redis Cache) be addressed and used from all execution models. The Managed Cache Service is still offered only for the sake of compatibility with older applications. Redis Cache should be selected for new projects.

➔ https://azure.microsoft.com/en-ie/documentation/services/redis-cache/#managed

3.4.3 In-Role Cache

The In-Role Cache transfers caching functionality to the instances of a cloud service itself, i.e., the memory resources used for caching are located in the VM instances used for application execution (whereby a percentage of resources are reserved for caching) or in VM instances used specially for caching purposes. The caching framework ensures that the memory content of the caching instances synchronizes among each other. If the application is configured accordingly, this cache can also be used for caching a user session.

➔ https://azure.microsoft.com/en-ie/documentation/services/redis-cache/#in-role

3.5 Hybrid data storage with StorSimple

StorSimple is a hybrid storage solution in which an appliance operated in the customer's data center provides local data storage. The stored data is replicated (encrypted if desired) in Microsoft Azure Storage. The security mechanisms in Azure Storage (3x storage, georeplication, etc.) ensure that the data is securely retained, eliminating the need for an additional backup.

➔ http://www.microsoft.com/en-ie/server-cloud/products/storsimple/

4 Analytics and Big Data Services

4.1 Event Hubs

In the context of the "Internet of Things," data streams from various sources such as Web sites, Apps, sensors, and other devices often need to be processed in real-time. Event Hubs is a high-grade, scalable data hub that can process millions of events per second. Messages are recorded per data source and cached in a partitioned storage area. This storage then enables the one-time or repeated processing of data via recipients connected in parallel.

➔ http://azure.microsoft.com/en-ie/services/event-hubs/

4.2 Stream Analytics

The analysis and evaluation of data stored in fixed memory ("resting") can be performed via HDInsight, for example. However, IoT scenarios often require the evaluation of data streams, i.e., data "in motion." Stream Analytics covers precisely this requirement, enabling analysis of data streams in real-time, i.e., data streams can be monitored, and abnormalities in the data can be detected and reported during operation. Queries on the data streams can be formulated in an SQL-based language. Database specialists therefore have a familiar technology for analyzing data streams.

➔ http://azure.microsoft.com/en-ie/services/stream-analytics/

4.3 Data Factory

Data processing in today's companies is marked by heterogeneous data storage (SQL, NoSQL, unstructured data, etc.) and processing components (databases, Big Data processors, etc.). Data in a company often passes through complex paths from generation or receipt of the data, through various data processing components, to storage or distribution of the data to various recipients. With Data Factory, local data such as that from SQL Server can be processed together with cloud-related data from Azure SQL Database, Blobs, and Tables. These data processing streams can be created, processed, and monitored through simple, highly available data pipelines. Data sources and data recipients can be defined, and the movement of the data in the company can be traced and monitored from a central location.

➔ http://azure.microsoft.com/en-ie/services/data-factory/

4.4 HDInsight

In addition to pure data storage, Microsoft Azure also provides capabilities for evaluation and processing of data stored in Azure. The flexible, straightforward provision of even major compute and storage resources, which can be deactivated after data is evaluated, makes Azure the ideal environment for such additional data processing.

HD Insight enables the analysis of large data sets with the aid of Hadoop, an open source framework very popular for calculating "Big Data problems," i.e., evaluations of data sets that cannot be stored in a relational database (e.g., because they are too large or they do not have a relational structure). In these situations, Microsoft Azure makes it possible to automatically provision Hadoop VMs, distributed data and computation algorithms, etc. In addition to efficiently computing MapReduce algorithms, it also supports other concepts such as Hive[3] and Pig[4].

➔ http://azure.microsoft.com/en-ie/services/hdinsight/

4.5 Machine Learning

Using Machine Learning, forecasts for further data can be made on the basis of existing data sets. Existing data can be divided into training data and control data. Forecast models can be created from pre-built analysis algorithms or analysis algorithms written using the R programming language. These can be trained using the training data. The forecasting accuracy achieved with the models can then be verified using the control data. The models can, in turn, be provided as a Web service. All of this is possible in an integrated graphical development environment where dataflow charts of data and process components can be created and configured by drag & drop.

➔ http://azure.microsoft.com/en-ie/services/machine-learning/

[3] Hive is a Data Warehouse infrastructure that supports data aggregation and related queries.

[4] Pig is a framework for describing and executing parallel computations.

4.6 Azure Search

Azure Search offers a search engine as a service. Various data sources can be configured as the basis for the search. The underlying evaluation models can be adapted and the search results thus tailored to business objectives. The search index is automatically created and updated.

http://azure.microsoft.com/en-ie/services/search/

4.7 SQL Data Warehouse

With SQL Data Warehouse, Data Warehouse solutions can be set up flexibly and their resources can be adapted to the specific data evaluation requirements. Compute and storage services can be scaled independently of one another, allowing the compute resources to be extended, reduced, or suspended while retaining the storage systems (for relational data or non-relational Blob storage).

http://azure.microsoft.com/en-ie/services/sql-data-warehouse/

4.8 Azure Data Catalog

The Azure Data Catalog is a directory service in which data sources in the company can be registered and made searchable. Different employees can register, describe, and assign metadata to data sources and use the data sources from their familiar data processing tools via Data Catalog.

http://azure.microsoft.com/en-ie/services/data-catalog/

5 Security of Azure-based Applications

Transferring data and application logic from the customer's data center to the cloud creates special security requirements for everyone involved. The systems operated in the cloud need to be protected against unauthorized access. Via the Microsoft Azure Trust Center, Microsoft provides information about the operation, data protection, security, and compliance of the underlying infrastructure, provides information for the creation of secure Azure-based applications, and with Microsoft Azure Active Directory it provides central building blocks for securing custom applications in the cloud.

5.1 Azure Active Directory

Microsoft Azure Active Directory is a directory service in which users and their organizational affiliations can be stored. Users can log in using this service. They will then receive a security token that they can pass to applications verify their user identity. Azure also allows synchronization of content with a locally operated Windows Server Active Directory. The Microsoft Azure Active Directory is primarily designed for use with cloud-based applications (such as Office 365). In scenarios requiring a full Active Directory (e.g., if a customer's data center is to be expanded to the cloud using Virtual Machines and Virtual Network), a virtual machine with a conventional Windows Server Active Directory should be used.

The Microsoft Azure Active Directory Graph provides a RESTful API for access to the Microsoft Azure Active Directory. This makes it possible to read a user's organizational affiliations and linkages with other users, the Social Graph.

With Active Directory Access Control, external identity providers can be used for user authentication in custom applications. This service is thus also well suited if users are not to be stored in the application itself (e.g., via Microsoft Azure Active Directory). A benefit for users is that they can log into the application using existing credentials (e.g., Windows Live ID with associated password). Fig. 5-1 illustrates the mechanism.

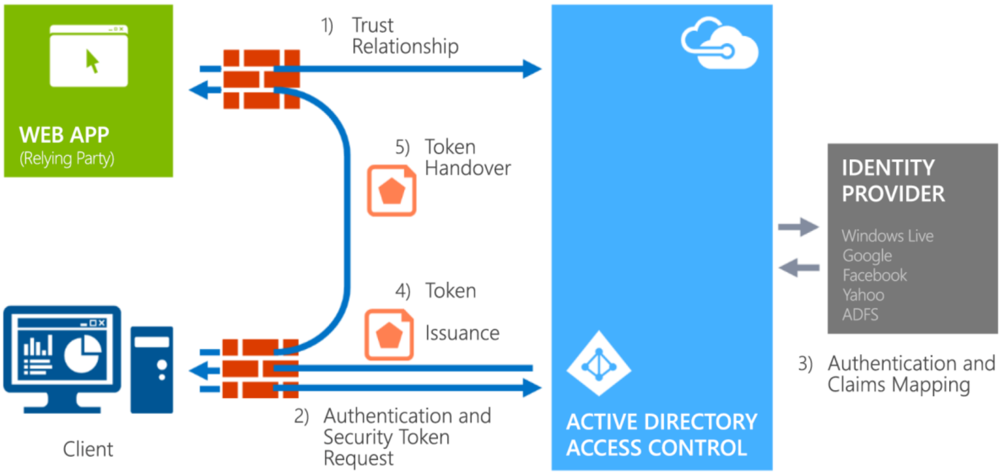


Fig. 5-1 Microsoft Azure Active Directory Access Control

The secured application establishes (by certificate exchange) a trust relationship with the Active Directory (step 1). It determines the generally supported identity providers through which users can log on. When opening the application, the user is able to select from these defined identity providers. Once the user has successfully authenticated (2), Access Control can convert profile information supplied by the provider into a token in a uniform format (e.g., Simple Web Token or SAML token) (3) and pass it to the user (4). The user can then pass this token to the application (5). The application then only has to evaluate tokens in the relevant format but does not have to interact with different identity providers. The Azure Active Directory handles this for the application. The following identity providers are supported: Microsoft Account, Google ID, Yahoo ID, Facebook ID, and Active Directory Federation Services.

➔ http://azure.microsoft.com/en-ie/services/active-directory/

5.2 Multi-factor authentication

Software applications with high security standards often use multi-factor authentication to authenticate users. This involves using more than one mechanism to verify a user. Such an authentication mechanism can be implemented using Microsoft Azure Active Authentication. In addition to password-based authentication, a mobile App, a telephone call, or an SMS can be used. Once this is configured for a user, the user has to save corresponding cellular phone numbers upon the next logon in Active Directory.

➔ http://azure.microsoft.com/en-ie/services/multi-factor-authentication/

5.3 Key Vault

Encryption is a proven means of securing data in the cloud. Secure management of the keys used is essential to effectively protecting data in the cloud. With Key Vault, keys and secret keys such as passwords can be encrypted using keys that are stored in HSMs (hardware security modules). Additional security is ensured if user keys are imported or generated in HSMs that are certified for FIPS 140-2 Level 2 and Common Criteria EAL4+. This way, the keys do not leave the HSM system. With Key Vault, keys are invisible to Microsoft and cannot be extracted.

➔ http://azure.microsoft.com/en-ie/services/key-vault/

5.4 Microsoft Azure Trust Center

The Microsoft Azure Trust Center provides a collection of information on the secure operation of Microsoft Azure as well as the applications executed and data stored on it. This includes descriptions of processes and standards that ensure security and data protection, as well as corresponding certificates that confirm this independently.

➔ http://azure.microsoft.com/en-ie/support/trust-center/

6 Media Processing and Distribution

6.1 Media Services

The flexible and (where necessary) worldwide provision of digital content is a major strength of Microsoft Azure. This makes it possible to store and convert audio and video content and flexibly deliver it to various devices. Microsoft Azure includes Microsoft Azure Media Services, which can handle such tasks centrally, for this purpose.

The provided functions include:

- Uploading media data (that is then stored in Blob storage)
- Coding and converting videos into various video and audio formats
- Access protection to media data for Digital Rights Management (DRM)
- Displaying advertising in video streams
- Streaming videos (optimized for various devices), scheduled if necessary

As with the other Azure Services, Media Services can also work with application components that are not necessarily executed on Microsoft Azure. They can still be called via RESTful interfaces.

Microsoft Azure CDN or third-party caching services can be used for optimum delivery of video content. Optimization to the consuming devices (e.g., Windows PC, Macintosh, HTML 5, iOS, Android, Windows Phone) and their respective playback options (resolution, screen size, etc.) is also possible.

➔ http://azure.microsoft.com/en-ie/services/media-services/

6.2 Content Delivery Network (CDN)

The Content Delivery Network (CDN) is a network of Microsoft Azure data centers distributed around the world. Content from Microsoft Azure Blob storage or Web Apps can be cached in the CDN for a certain period of time, ensuring efficient delivery of Blob storage content: a user is delivered content from the nearest data center when accessing Blob storage. On first access by a user, Azure checks whether the requested data is already available in the CDN location. If this is the case, the data is delivered from that location. If the data is not yet there, it is delivered from the original from Blob storage. A copy is stored in the CDN so that subsequent calls can be served from there.

➔ http://azure.microsoft.com/en-ie/services/cdn/

7 Integration of Azure-based Applications into In-house IT

In many software projects, the question arises as to how cloud resources can be linked with local IT. It is not realistic that a company's IT migrate entirely to the cloud, nor would it make sense to go without the benefits of the cloud (flexible provision of IT, usage-based billing, high scalability, etc.). Microsoft Azure provides various ways of linking Azure-based resources with local resources. At the application level, there is the Service Bus; SQL Azure DataSync can be used at the data level; and at the network level, Virtual Network is an option provided by the Traffic Manager, supplemented by a Routing Service.

7.1 Data integration via SQL DataSync

The SQL Azure DataSync Service is a cloud service in SQL Azure that makes it possible to synchronize Azure-based SQL databases among one another as well as with locally operated SQL server databases. Synchronization can be set up and administered through a graphical interface. The synchronization frequency, data transfer direction (bidirectional, only upload, only download), strategies for conflict resolution, and the content to be synchronized (databases, tables, columns, and table rows (filtered by field values)) can be set.

Fig. 7-1 outlines how it operates. Data transfer between the involved databases is handled via a DataSync hub. The DataSync hub must be an SQL Azure database. A locally operated database cannot be used as a hub.

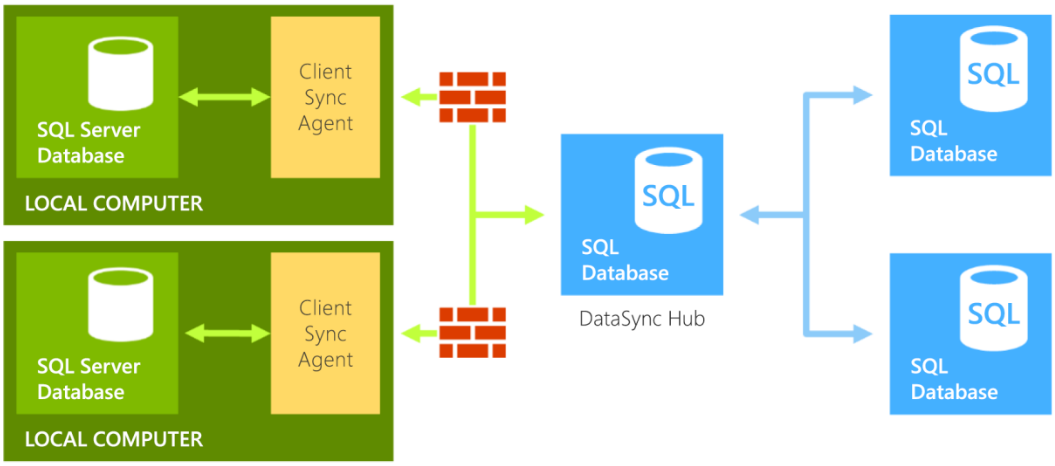


Fig. 7-1 Microsoft Azure SQL DataSync

A possible usage scenario for DataSync is the synchronization of data to regional offices. In this scenario, DataSync contains all the data, and regional offices synchronize only the data specifically for them.

➔ http://msdn.microsoft.com/en-ie/library/windowsazure/hh456371.aspx

7.2 Application integration

In shared applications, the components that should interact with one another must have a means of exchanging messages. Depending on the type of distribution, this exchange must also be done across company and firewall boundaries. This is a usage case particularly in hybrid scenarios in which Azure-based components should interact with services in a locally operated data center (which is usually protected by firewalls, etc. against access "from the outside"). Microsoft Azure provides mechanisms for developers to implement such message exchange scenarios: the Queue Service and the Service Bus.

7.2.1 Queue Service

With the Queue Service, the Microsoft Azure platform offers a messaging service through which communication partners can communicate with one another asynchronously by exchanging messages. Access is made over http/https. APIs are available for the main programming languages and technologies such as .NET, Java, and PHP for convenient access to queues and messages. The use of the Queue Service is not limited to use within an Azure-based application. Communication partners can also be outside the cloud. Fig. 7-2 outlines how the Queue Service operates.

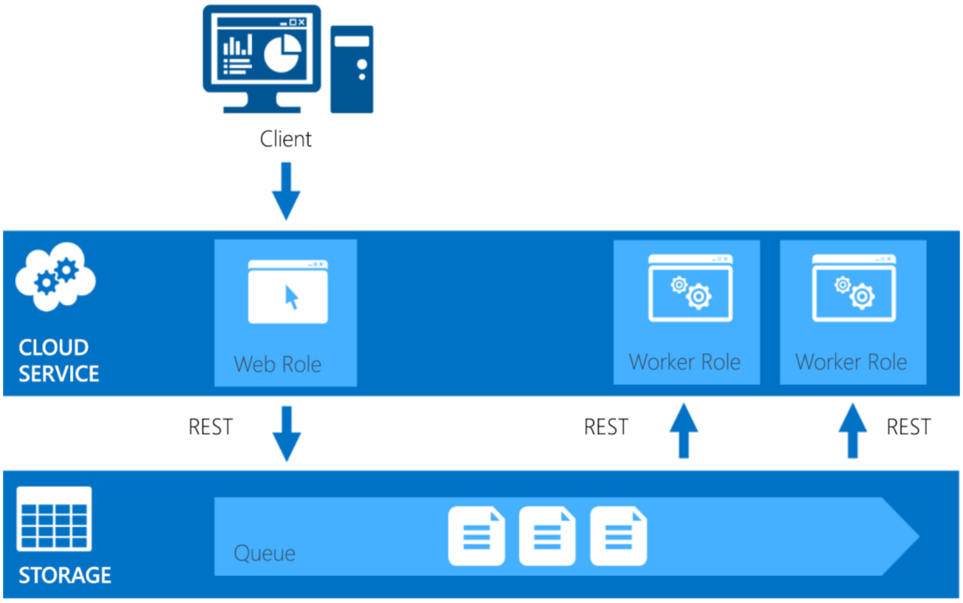


Fig. 7-2 Microsoft Azure queues

The most common usage scenario for the Queue Service is the exchange of messages between individual components of a cloud service. A frontend component (e.g., a Web role) accepts work orders from the user and places these as messages into a queue from which they are then read and processed by one or more backend components (e.g., worker role instances). Such an application can be scaled easily (e.g., based on the fill level of the queue): if the queue is often only slightly full or is empty, it may be possible to reduce the number of backend components. If the queue is always heavily filled, it may be possible to process messages more quickly by adding further backend components.

➔ https://azure.microsoft.com/en-ie/documentation/articles/storage-dotnet-how-to-use-queues/

7.2.2 Service Bus

The Microsoft Azure Service Bus provides a communication and integration platform for networking distributed Web services. Simple and complex message streams can be implemented regardless of where the individual communication partners are executed (in the cloud or in the customer's data center). Support is provided for point-to-point connections (unidirectional and bidirectional), queues for caching exchanged messages, as well as publish-subscribe patterns about topics where multiple senders can dispatch messages to multiple recipients.

➔ http://azure.microsoft.com/en-ie/services/service-bus/

7.2.2.1 Service Bus Relay

A simple interaction scenario is presented in Fig. 7-3: a challenge for the Azure App, which wants to call a locally executed Web service, is that the Web service is protected against external access by an upstream firewall. The Service Bus, however, allows a communication relationship to be established. To do so, the Web site registers with the Service Bus (step 1), which configures an endpoint for the Web service. The Azure App can now call this endpoint instead of the Web service (2). The Service Bus inspects the call and can forward it to the Web service on the connection established to the Web service in step 1.

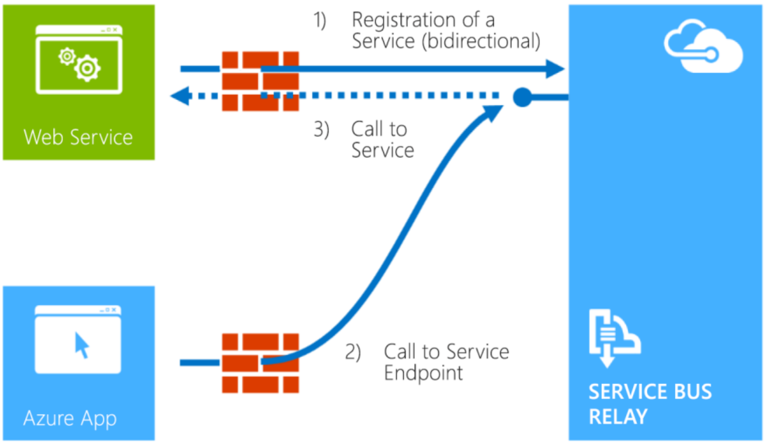

Web Service
1) Registration of a Service (bidirectional)
3) Call to Service
Azure App
2) Call to Service Endpoint
SERVICE BUS RELAY

Fig. 7-3 Microsoft Azure Service Bus

7.2.2.2 Service Bus queue

The relay requires that the registered recipient actually be online, i.e., can receive the messages. If the user is offline, the sender will immediately receive an error message and messages cannot be sent. Queues can be defined to permit message dispatch in such scenarios. Senders can then dispatch messages in any case. A recipient then reads the accumulated messages once going online.

7.2.2.3 Service Bus topic

Topics with an endpoint to which senders can dispatch their messages can be configured in the Service Bus. Subscriptions can be defined for topics. Filter rules can be defined for each subscription and can be used to determine whether a message sent to a topic is to be copied into the queue for the subscription in question. Recipients can then read messages from the subscription queues.

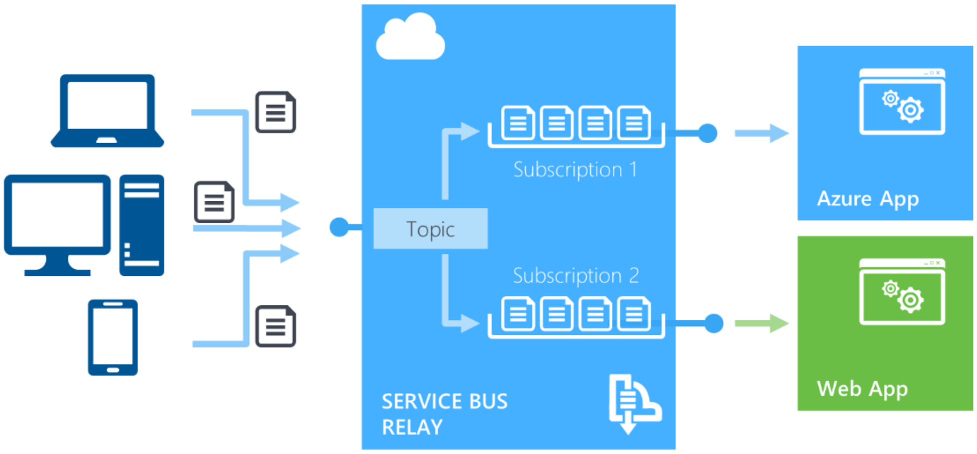


Fig. 7-4 Microsoft Azure Service Bus topics and subscriptions

A usage scenario for this is application monitoring. Instead of writing status messages to a log file, these notifications could be sent as a message to a Service Bus topic. Depending on the notification type, the recipient can then write pure status information to a log file, but error messages could go into a separate subscription where the recipient can process them directly (e.g., sending an SMS to an administrator).

➔ http://msdn.microsoft.com/en-ie/library/windowsazure/ee732537.aspx

7.2.3 BizTalk Services

The Microsoft Azure platform contains BizTalk Services for more complex integration scenarios where existing backend systems from a local data center are to be integrated via the Service Bus. The Service Bus can be used as a connector for linking local IT systems (e.g., SAP, Oracle DB, Oracle EBS, Siebel, SQL DB) for interaction, while the exchanged messages can be filtered and converted into various notification formats. Notification-driven workflows can be defined.

➔ http://azure.microsoft.com/en-ie/services/biztalk-services/

8 Network Services

Microsoft Azure offers a range of network services that can be used to network Azure resources in three areas: networking virtual machines among one another, as well as their connections from the Internet, or to one or more local data centers. Fig. 8-1 shows the interaction of several network services.

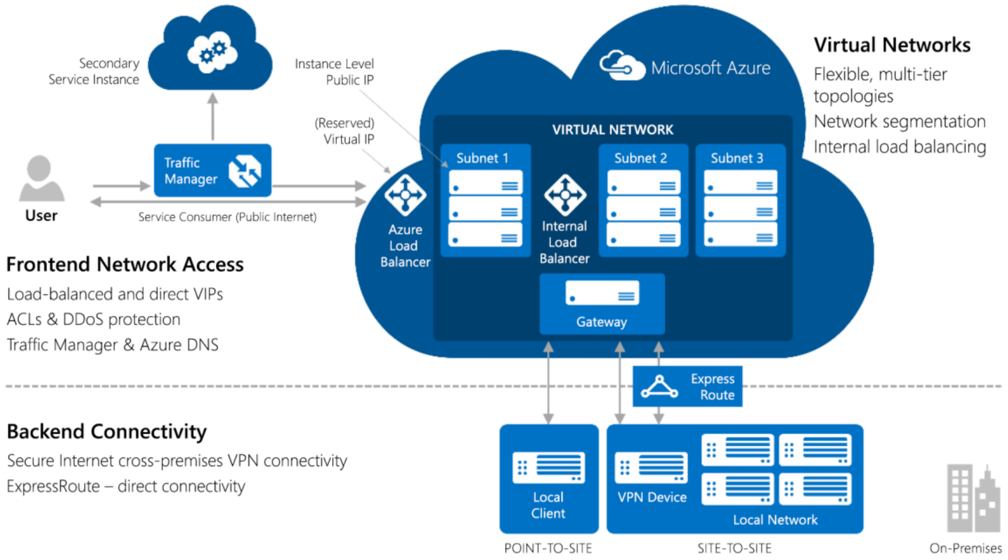


Fig. 8-1 Network services in Microsoft Azure

Virtual machines can be networked together via a Virtual Network. Subnetworks and reliable network traffic can also be defined. Network traffic from the Internet first encounters the Azure load balancer, which distributes the traffic to the Azure resources after performing various security checks. An internal load balancer can also be configured within a Virtual Network.

A Virtual Network can be securely connected to one or more local data centers via a gateway. A dedicated line with guaranteed bandwidth can also be used via ExpressRoute for this purpose. An Azure-based service can be installed with worldwide distribution to provide user groups with optimal service around the world. Calls can be distributed appropriately to the various service instances via Traffic Manager.

8.1 Virtual Network

With Virtual Network, Microsoft Azure can be used as an extension to local infrastructure where additional resources can be deployed flexibly and temporarily from the cloud. Network configurations can be created to integrate local computers or networks with network nodes in Microsoft Azure into a virtual network. The cloud resources can be provided their own IPv4 addresses, and custom routing tables can be expanded to include these resources. Combined with Virtual Machines, this makes it possible to migrate virtual machines back and forth between the customer's data center and Microsoft Azure and to provide appropriate configurations for this at the network level so that the migration occurs entirely transparently for the other network elements. Ultimately, users can utilize Azure-based resources as though they were in their own data center.

On the local side, the integration can be performed via a dedicated Windows Server machine that manages the connection to the cloud resources. But this is not limited solely to Windows machines. Dedicated network hardware can be used for the integration itself; the hardware must be configured appropriately by a network administrator. Operating systems other than Windows can therefore be used in the relevant network nodes.

Through the Virtual Network, connections to one or more local sites (e.g., multiple offices of a company) or between virtual networks in Azure can be set up from within Microsoft Azure.

➔ http://azure.microsoft.com/en-ie/services/virtual-network/

8.2 ExpressRoute

The connection path from a client to Microsoft Azure normally includes a certain section on the public Internet. This results in access times being influenced by other Internet traffic. To avoid this, a private network connection can be set up from Azure data centers to the customer's local compute resources via ExpressRoute. This can significantly increase the throughput, stability, and reliability of connections to Azure.

➔ http://azure.microsoft.com/en-ie/services/expressroute/

8.3 Traffic Manager

The Microsoft Azure Traffic Manager is a service that forwards requests sent to an Azure Service on the basis of freely definable policies and availabilities to service instances that can be installed worldwide in Azure or in external environments. Three types of forwarding policies can be defined: Performance, Failover, and Round Robin. A policy contains a DNS name. The Traffic Manager replies to requests to this name with the IP address of the next available service instance that fulfills this policy.

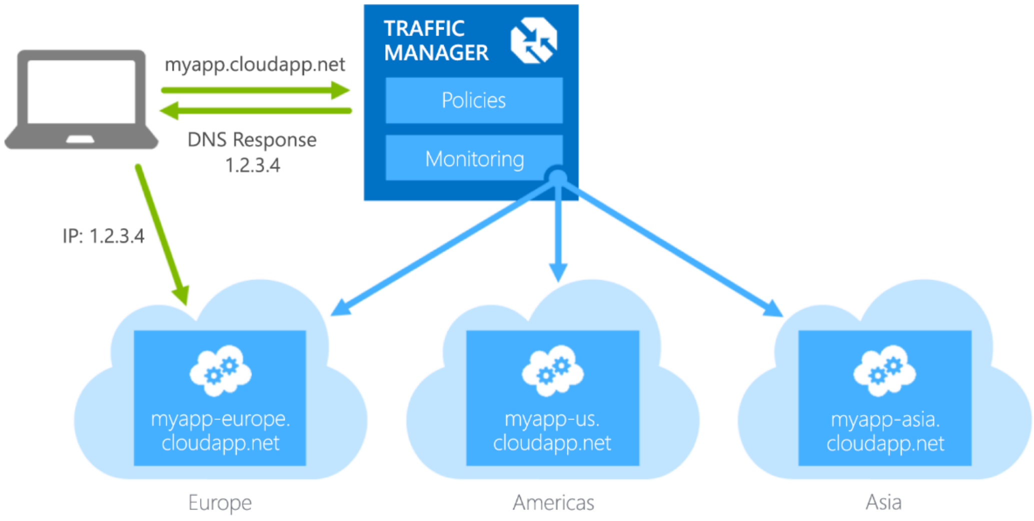


Fig. 8-2 Microsoft Azure Traffic Manager

The decision of which instance a request is forwarded to is thus based both on the policy and the availability. This makes it possible to distribute the instances of a service worldwide, to normally direct users to the nearest instance, and in case this instance fails, to go to the second-nearest instance, etc. Due to support for including instances operated outside of Azure, failover scenarios can be implemented where access is first made to on-premises instances and, in case of an error, instances in the cloud are accessed.

➔ http://azure.microsoft.com/en-ie/services/traffic-manager/

© 2016, Microsoft Deutschland GmbH

8.4 Load Balancer

All data traffic from the public Internet that aims to access Azure resources must pass through the Azure load balancer, which processes various protocols (e.g., http, https, smtp) and where appropriate forwards calls to the required Azure resources after performing security checks. The load balancer checks the availability of the resources on a regular basis and ensures that calls are sent only to available resources. For multilayered applications, within a Virtual Network an internal load balancer can also be configured between the individual layers.

http://azure.microsoft.com/en-ie/services/load-balancer/

8.5 DNS Service

DNS domains can be hosted in Azure using Azure DNS. The service uses a global network of name servers to provide high performance for requests.

http://azure.microsoft.com/en-ie/services/dns/

8.6 VPN Gateway

A VPN gateway can be configured to securely connect a Virtual Network to local resources. The VPN gateway enables secure connection to a local computer (point-to-site) or to one or more local networks (site-to-site). An appropriate gateway is also required on the local side in case of site-to-site connections.

http://azure.microsoft.com/en-ie/services/vpn-gateway/

8.7 Application Gateway

For Web applications, the Application Gateway can provide useful routing and load balancing services at the application level, permitting the simple configuration of sticky sessions (the results of calls from a particular client are always directed to the same server) via the Gateway. SSL offloading can be implemented via the gateway, i.e., the gateway handles computationally intensive tasks to decrypt requests and encrypt replies in SSL-based communication, which relieves the Web server from performing these tasks itself.

http://azure.microsoft.com/en-ie/services/application-gateway/

9 DevOps with Microsoft Azure

9.1 Services for software development

9.1.1 Visual Studio Online

Visual Studio Online (VSO) provides functions as cloud services that were previously available only on the server components of a Visual Studio-based development environment, i.e., Team Foundation Server. With VSO, development projects can be planned, created, and distributed without having to configure server environments for this purpose. At the push of a button, VSO creates team projects in which source code can be managed centrally via Team Foundation Version Control (TFVC) or in a distributed manner via Git. Work tasks, functions, bugs, etc. can be managed via backlogs and scheduled, monitored, and traced using various process templates (Scrum, Kanban, agile processes). VSO can also handle the build process, automated testing, and provision.

➔ http://azure.microsoft.com/en-ie/services/visual-studio-online/

9.1.2 Application Insights

With Application Insights, correspondingly configured applications from various platforms can be monitored at runtime and their runtime behavior evaluated. Various metrics (performance, faults, etc.) can be controlled and tracked. The telemetry data supplied by Application Insights from the monitored applications can be processed, visualized, filtered, and segmented to obtain a constant overview of the state and usage of a monitored application. Applications in various languages (C#, VB, JavaScript, Java, Ruby, PHP, Python, etc.) and platforms (ASP.NET, Android, iOS, Cordova, Angular, Node.js, Joomla, SharePoint, etc.) as well as numerous logging frameworks (Log4Net, NLog, Java, Log4J, etc.) are supported.

➔ http://azure.microsoft.com/en-ie/services/application-insights/

9.2 Services for system operations

9.2.1 Operational Insights

As part of the Microsoft Cloud Management Suite, Operational Insights makes it possible to collect, store, and analyze logging data from almost any Windows or Linux-based servers and services in the customer's data center or in the cloud. The collected machine data can be managed, combined, and visualized to provide a basis for making decisions on system operations, such as capacity management, maintenance, and configuration of systems.

➔ http://azure.microsoft.com/en-ie/services/operational-insights/

9.2.2 Data transfer and data backup

The storage systems in Microsoft Azure are designed for high availability. Redundant storage and (optional) replication to other data centers ensure that data loss practically cannot occur. Azure is therefore excellently suited as a backup medium, even for data normally kept locally. Backup and recovery services allow the backup of server systems in Azure.

9.2.2.1 Import/Export Service

The transfer of large amounts of data over the Internet to Blob storage can be time-consuming depending on the available bandwidth. With sufficiently large amounts of data, it is more efficient to put the data on hard disks and send them via a package service to an Azure data center, where their content will then be copied to a storage account. Such data transfers can be configured using the Azure Import/Export Service. For data export, empty hard disks can be sent to a data center, where the content of a storage account will be loaded onto them. The hard disk data can be encrypted both during import and export.

➔ http://azure.microsoft.com/en-ie/pricing/details/storage-import-export/

9.2.2.2 Backup Service

The backup tools in Windows Server 2012, Windows Server 2012 Essentials, and System Center 2012 Data Protection Manager, working together with the Microsoft Azure Backup Service, allow the creation of (incremental or full) backups of local systems in Microsoft Azure Storage. Backups can then be loaded very easily on any server systems as needed.

➔ http://azure.microsoft.com/en-ie/services/backup/

9.2.2.3 Site Recovery

The Hyper-V Recovery Manager enables the coordinated replication and the recovery of resources in a private cloud environment in a second environment. Using System Center 2012 Virtual Machine Manager, private clouds can be secured by continuous replication to a second data center location and, where necessary (in case of failure of the first location), activated.

http://azure.microsoft.com/en-ie/services/site-recovery/

9.2.3 Scheduler

Actions can be executed according to any desired schedules using the Scheduler, allowing http/s endpoints to be called or messages placed in a storage queue, which application logic (e.g., in a Web service or in a cloud server worker role) can then respond to. This makes it possible to trigger actions occurring in the future on a regular basis or a single time.

http://azure.microsoft.com/en-ie/services/scheduler/

9.2.4 Automation

Automation greatly facilitates the management of Azure resources by enabling the automation of tasks such as the creation of resources, their deployment, monitoring, and administration as part of a workflow-based engine. Associated resources (e.g., a cloud service with associated storage account and Active Directory setup) can be created and managed using what are referred to as "runbooks." The individual resources then no longer have to be created individually and separately.

http://azure.microsoft.com/en-ie/services/automation/

10 Getting Started with Microsoft Azure

10.1 Blogs and Web sites

Access to Microsoft Azure

- Free demo account:
  https://azure.microsoft.com/en-ie/free/
- Access to Microsoft Azure through an MSDN subscription:
  https://azure.microsoft.com/en-ie/pricing/member-offers/msdn-benefits/
- Access to Microsoft Azure through BizSpark:
  http://azure.microsoft.com/en-ie/offers/ms-azr-0064p/

Tools and Software Development Kits (SDKs)

- Software Development Kits (SDKs) for .NET, Node.js, PHP, Java, Python, Ruby:
  http://azure.microsoft.com/en-ie/downloads/
- Windows Azure PowerShell:
  http://go.microsoft.com/?linkid=9811175&clcid=0x407
- Virtual Machine Assessment Tool:
  http://azure.microsoft.com/en-ie/downloads/vm-readiness-assessment/
- Assistant for migration of Web sites to Web Apps:
  http://go.microsoft.com/?linkid=9863189&clcid=0x407

Documentation (whitepapers, videos, etc.)

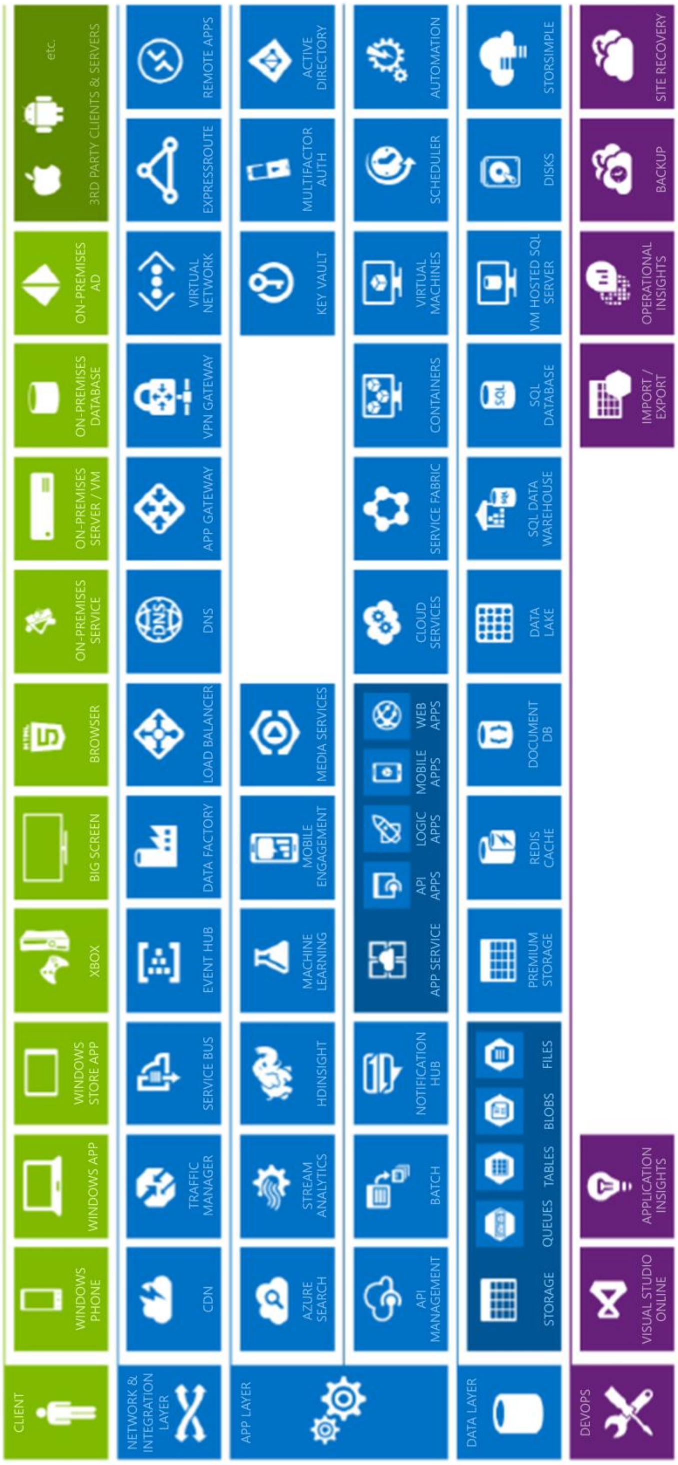
- Microsoft Azure Developer Center:
  http://azure.microsoft.com/en-ie/develop/net/
- Technical documentation for Microsoft Azure on MSDN:
  https://msdn.microsoft.com/en-ie/library/azure/dn578280.aspx
- Official Microsoft Azure blog:
  http://azure.microsoft.com/en-ie/blog/

Annex A: Overview of Azure Services

Azure Service (English title)	**Azure Service (German title)**	**Description**	**Chapter**
Azure Compute: Execution of custom application logic			
App Service	App-Dienst		2.1
Web Apps	Web-Apps	Hosting of Web applications	2.1.1
Mobile Apps	Mobile Apps	Backend services for Mobile Apps	2.1.2
Logic Apps	Logic Apps	API-based workflows	2.1.3
API Apps	API Apps	Hosting of Web-based interfaces	2.1.4
Cloud Services	Cloud Services	Hosting of multilayered Apps (PaaS)	2.2
Service Fabric	Service Fabric	Microservices-based applications	2.3
Virtual Machines	Virtuelle Computer	Hosting of virtual machines	2.4
Batch	Batch	Execution of batch processes	2.5
Mobile Engagement	Mobile Engagement	Control and analyses of Mobile Apps	2.6
Notification Hubs	Benachrichtigungs Hub	Push notifications to many clients	2.7
RemoteApp	RemoteApp	Provisioning of Windows Apps	2.8
API Management	API Management	Distribution and management of APIs	2.9
Azure Data Services: Storage of data			
SQL Database	SQL Datenbank	Relational database-as-a-service	3.1.2
DocumentDB	DocumentDB	Storage of JSON-based objects	3.2.1
Storage	Speicher		
Tables	Tables	NoSQL database (key entity)	3.2.2
Blobs	Blobs	Storage of large binary data	3.2.3
Files	Files	SMB shares for VMs and Cloud Services	3.3.1
Disks	Disks	Persistent file system for VMs	3.3.2
Redis Cache	Redis Cache	Central cache based on Redis	3.4.1
Managed Cache	Verwalteter Cache	Centrally provisioned cache	3.4.2
In-Role Cache	Rollen-bezogener Cache	Cache within Cloud Service VMs	3.4.3
StorSimple	StorSimple	Hybrid, secure data storage	3.5
Analytics and Big Data Services			
Event Hubs	Event Hubs	Processing of event messages	4.1
Stream Analytics	Stream Analytics	Evaluation of data streams	4.2
Data Factory	Data Factory	Management of data pipelines	4.3
HDInsight	HDInsight	Hadoop-as-a-Service	4.4
Machine Learning	Machine Learning	Creation of forecast models	4.5
Azure Search	Azure Search	Search-engine-as-a-service	4.6

Azure Service (English title)	Azure Service (German title)	Description	Chapter
SQL Data Warehouse	SQL Data Warehouse	Elastic data-warehouse-as-a-service	4.7
Azure Data Catalog	Azure Data Catalog	Catalog of data sources in the company	4.8
Security of Azure-based Applications			
Azure Active Directory	Azure Active Directory	Identity store and token service	5.1
Multi-Factor Auth	Multi-Faktor Auth	Multi-factor authentication	5.2
Key Vault	Key Vault	Secure key management	5.3
Application services for custom applications			
Media Services	Media Services	Media workflows in the cloud	6.1
Content Delivery Network	Content Delivery Network	Global caching of static content	6.2
Integration of Azure-based Applications into In-house IT			
SQL DataSync	SQL DataSync	Synchronization of SQL DBs	7.1
Queues	Warteschlangen	Message dispatch	7.2.1
Service Bus	Service Bus	Internet Service Bus	7.2.2
BizTalk Services	BizTalk Services	Integration of backend systems	7.2.3
Network Services			
Virtual Network	Virtual Network	Network-to-network VPN	8.1
ExpressRoute	ExpressRoute	Exclusive network connections	8.2
Traffic Manager	Traffic Manager	Global routing of service access	8.3
Load Balancer	Load Balancer	Load balancer for groups of VMs	8.4
DNS Service	DNS Service	Hosting a DNS domain	8.5
VPN Gateway	VPN Gateway	Cross-location connections	8.6
Application Gateway	Application Gateway	Web routing and load balancing services	8.7
DevOps with Microsoft Azure			
Visual Studio Online	Visual Studio Online	Managing development projects	9.1.1
Application Insights	Application Insights	Information on app runtime behavior	9.1.2
Operational Insights	Operational Insights	Information on system operations	9.2.1
Import/Export	Import/Export	Data transfer in Blobs via hard disks	9.2.2.1
Backup	Backup	Backing up virtual machines	9.2.2.2
Site Recovery	Site Recovery	Recovering virtual machines	9.2.2.3
Automation	Automatisierung	Automated management	9.2.3

Annex B: Structure of the platform



Index

About the author

Holger Sirtl has been a Senior Technical Evangelist at Microsoft in Munich since 2006, and in this role he advises companies on setting up cloud-based application architectures. His work focuses on cloud computing and the Microsoft Azure platform. Before joining Microsoft, Holger Sirtl worked for six years as a technology consultant for a leading international consultancy and for two years as a Senior IT Project Manager for a major German energy utility.

Address

Holger Sirtl
Microsoft Deutschland GmbH
Konrad-Zuse-Straße 1
85716 Unterschleißheim, Germany

Online

Blog:	http://blogs.msdn.com/hsirtl
Twitter:	@hsirtl
Xing:	https://www.xing.com/profile/Holger_Sirtl
LinkedIn:	http://de.linkedin.com/in/hsirtl

© 2016, Microsoft Deutschland GmbH