

# Cloud Computing mit Windows Azure

Ruben Deyhle

Universität Ulm, Abt. DBIS  
ruben.deyhle@uni-ulm.de

**Zusammenfassung** Riesige, skalierende Webanwendungen, die mit Millionen von Nutzern funktionieren, sind heute kein Problem mehr – dank „Cloud Computing“. Es haben sich mittlerweile mehrere Anbieter von Cloud-Computing-Plattformen etabliert, welche die Entwicklung von Cloud-basierten Webanwendungen relativ einfach ermöglichen. Eine besonders interessante Plattform stellt dabei Windows Azure von Microsoft dar, das es Entwicklern mit bekannten Entwicklungstools ermöglicht, ihre Webanwendung in die Cloud zu bringen. Diese Arbeit liefert einen kurzen Überblick über Cloud Computing und beschäftigt sich dann ausführlich mit Windows Azure. In einem praktischen Teil wird die Funktionsweise von Windows Azure anschaulich erklärt.

## 1 Einführung

„Cloud Computing“ ist seit einigen Jahren ein häufig anzutreffendes Schlagwort in der IT-Welt. Dabei handelt es sich – wie so oft – um einen Begriff, der für viele verschiedene Sachverhalte genutzt werden kann; eine klare Definition ist gar nicht so einfach zu finden. Oft werden darunter einfach Online-Dienste verstanden; fast jede Webanwendung lässt sich prinzipiell dem Feld des Cloud Computing zuordnen: stets läuft auf Serverseite „in der Cloud“ Software, die nach Bedarf praktisch unabhängig vom Standort und des lokalen Systems genutzt werden kann.

Bekannt gemacht haben „Cloud Computing“ Dienste wie Google Docs oder Dropbox – Webanwendungen für den Enduser, die alle eine Gemeinsamkeit haben: die Daten sind stets online gespeichert, dadurch immer synchron und von einer Vielzahl von Geräten aus abrufbar. Die Hard- und Softwareausstattung des Endgeräts ist dabei fast völlig irrelevant, da es nur der Nutzerinteraktion dient. Die Software selbst läuft auf Servern „in der Cloud“, wo ebenfalls auf notwendige Hardwareinfrastruktur zurückgegriffen werden kann. Bei Google Docs beispielsweise wird keine teure Officesoftware benötigt, die auf dem eigenen Computer installiert werden muss und mehrere GB Speicherplatz belegt, sondern über den Browser steht dem Nutzer eine vollwertige Office-Suite zur Verfügung, sobald er sie braucht.

Und genau dies ist der allen dem „Cloud Computing“ zugehörigen Systemen

gemeinsame Aspekt: Anwendungen oder Dienste können nach Bedarf angefordert werden, während „hinter den Wolken“ die Rechenzentren und Plattformen stehen, die die entsprechende Leistung bieten [1].

Neben den erwähnten Diensten, die konkrete Anwendungen bereitstellen (Software as a Service, SaaS), sind vor allem Cloud-Computing-Dienste interessant, die generischere Dienste bieten (Platform as a Service, PaaS) – oder gleich eine ganze IT-Infrastruktur auf Cloud-Basis (Infrastructure as a Service, IaaS). Jedoch ist die Unterscheidung gerade zwischen IaaS (low-level) und PaaS (higher-level) schwierig [2].

Im Gegensatz zu SaaS-Diensten richten sich PaaS-Dienste an Entwickler, die die Cloud-Plattformen für eigene Software nutzen möchten – welche sich dann wiederum als SaaS dem Endnutzer zur Verfügung stellen lässt.

Doch nicht nur das, natürlich sind PaaS-Dienste auch für wissenschaftliche Zwecke interessant. Durch die bedarfsgerechte Abrechnung eröffnen sich Möglichkeiten, die mit klassischer Rechnerinfrastruktur niemals finanzierbar wären: wird für ein Experiment für einige Stunden massive Rechenleistung benötigt, kann diese einfach aus der Cloud „gemietet“ werden, anstatt selbst entsprechende Rechenleistung aufzubauen, die danach vielleicht nie wieder benötigt wird. Genauso kann oft ein großer Geschwindigkeitsvorteil erzielt werden: anstatt einen Rechner für 1000 Stunden arbeiten zu lassen, lässt man 1000 Cloud-Rechner nur 1 Stunde arbeiten [2].

Um einen Überblick über die aktuell wichtigen PaaS-Provider und typische Anwendungsfälle zu bekommen, werden diese nun zuerst kurz vorgestellt. Der Hauptteil wird sich dann konkret mit Microsofts PaaS-Dienst, Windows Azure, beschäftigen. Dabei wird versucht, einen umfassenden Überblick über Funktionsweise und Möglichkeiten zu vermitteln. Im Praxisteil wird die Einrichtung und Installation einer kleinen cloudbasierten Anwendung mit Windows Azure beschrieben. Schließlich wird Windows Azure in den Kontext seiner Mitbewerber gestellt und ein grober Vergleich angestellt. Außerdem wird die Frage beantwortet, in welchen Fällen sich der Einsatz von Cloud Computing und insbesondere von Windows Azure empfiehlt.

## **2 Marktübersicht**

In diesem Kapitel wird kurz auf die wichtigsten Anbieter von Cloud-Computing-Diensten eingegangen.

### **2.1 Google Apps und Google App Engine**

Google ist bekannt für seine kostenlosen, weit verbreiteten Dienste im Software-as-a-Service-Bereich – zum Beispiel Google Mail, Google Docs oder Google Calendar. Diese Dienste stehen als „Google Apps“ mit weitreichenden Konfigurations- und Personalisierungsmöglichkeiten auch Unternehmen und

professionellen Nutzern zur Verfügung. Zusätzlich zu den Apps von Google können über einen Marketplace zahlreiche weitere Apps bezogen und angeboten werden.<sup>1</sup>

Doch für die Google-Apps-Plattform entwickelte Anwendungen müssen nicht öffentlich im Marketplace angeboten werden, sie können auch nur intern genutzt werden – oder unabhängig von Google Apps. Diesen PaaS-Dienst nennt Google „Google App Engine“. Unterstützt wird Software in Python und in Java, entsprechende APIs stellt Google bereit. Auf die Datenbankkomponente „Datastore“ wird über die SQL-ähnliche GQL zugegriffen. Die Abrechnung erfolgt – Cloud-Computing-typisch – nach Rechenzeit und nach Bandbreitenausnutzung. Quotas, um die maximalen Kosten pro Tag zu beschränken, lassen sich konfigurieren [3].

## 2.2 Amazon Web Services

Amazon, bekannt als großer Online-Versandhändler, bietet bereits seit 2002 mit seinen Web Services (AWS) Clouddienste an. Ursprünglich wurde die eigene IT-Infrastruktur nur zur besseren Auslastung amerikanischen Unternehmen bereitgestellt, mittlerweile hat sich Amazon zu einem der größten Cloud-Computing-Anbieter entwickelt [1]. Dabei positioniert sich Amazon als reiner PaaS- bzw. IaaS-Anbieter, SaaS führt Amazon im Gegensatz zu Google nicht im Programm und wendet sich folglich auch nicht an Endanwender.

Stattdessen bietet Amazon mächtige Cloud-Instrumente an, am bekanntesten dürfte davon Amazon S3 sein, der „Simple Storage Service“. Er stellt als „Festplatte“ die Basis der AWS dar, die anderen Dienste bauen darauf auf. S3 wird auch von reinen SaaS-Anbietern genutzt, beispielsweise Dropbox [4].

Amazon Elastic Compute Cloud (EC2) ist der Dienst, der AWS zu einem Infrastructure-as-a-Service-Dienst macht. Es ist möglich, beliebige (virtuelle) Server-Instanzen nach Bedarf zu buchen. Man kann also nicht nur Software in die Cloud laden und dort laufen lassen, sondern es können sogar ganze Systeme in die Cloud verlagert werden.

Als Datenbank wird Amazon SimpleDB als drittes großes Standbein der AWS angeboten. Etliche weitere Dienste runden das Angebot ab. Diese nutzen alle das Cloud-Computing-typische Abrechnungsmodell, bei dem direkt nach genutzter Rechenleistung bzw. genutztem Datenvolumen abgerechnet wird [5].

## 2.3 Salesforce

Salesforce konzentriert sich wie Google auf Software as a Service und bietet in diesem Bereich zahlreiche Businessprodukte an, beispielsweise für Kontaktmanagement oder Marketing. Unter dem Namen „Force.com“ stellt Salesforce seine Platform-as-a-Service-Dienste zur Verfügung, die alle Bereiche von Softwareentwicklung und -betrieb abdecken. Als Sprache kommt das Java-ähnliche

<sup>1</sup> <http://www.google.com/enterprise/marketplace/>

„Apex“ zum Einsatz, zahlreiche APIs und Entwicklertools stehen zur Verfügung.

Das Preismodell weicht dabei vom Cloud-Gedanken ab: anstatt nach Verbrauch wird nach User und Monat abgerechnet [1,6].

## 2.4 Windows Azure

Microsofts Cloud-Computing-Ansatz firmiert unter dem Namen Windows Azure, hat allerdings nur wenig mit dem Windows-Betriebssystem zu tun. Eine ausführliche Analyse des Platform-as-a-Service-Dienstes folgt nun im Hauptteil.

# 3 Windows Azure

In diesem Kapitel wird der Aufbau und die Funktion von Windows Azure erklärt.

## 3.1 Charakterisierung

Microsofts Windows Azure verfolgt einen Platform-as-a-Service-Ansatz. Es stellt keine Webanwendungen für den Endnutzer oder Unternehmen bereit, sondern richtet sich an Entwickler und Unternehmen, die ihre eigene Software auf Windows Azure in der Cloud laufen lassen wollen.

Zwar möchte Microsoft auch eigene, Endanwender-orientierte Projekte mithilfe von Azure an den Start bringen, diese werden aber nicht unter dem Namen „Windows Azure“ firmieren, sondern nur die Technik von Azure nutzen. Auch bietet Microsoft (noch) keine Business-Software an, die über Windows Azure direkt genutzt werden könnte.

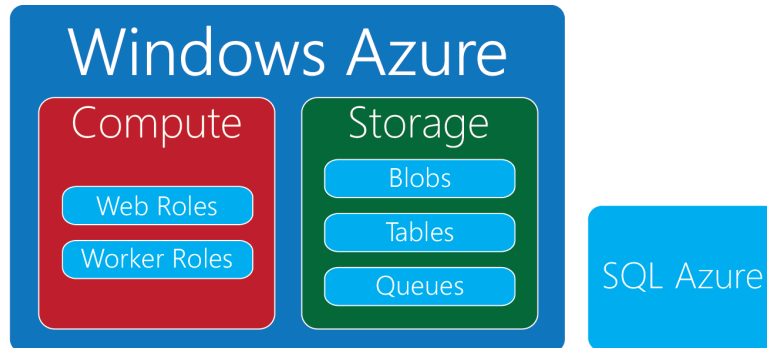
Stattdessen wird eine Plattform geboten, auf der praktisch beliebige cloudbasierte Webanwendungen jeglicher Größe betrieben werden können.

Die Abrechnung erfolgt grundsätzlich Cloud-typisch nach Nutzung. Dabei können Pakete gebucht werden, die gewisse Nutzungsvolumen beinhalten; was darüber hinaus geht, wird dann nach Verbrauch abgerechnet [7].

## 3.2 Aufbau

Die Windows-Azure-Cloud ist über viele Serverfarmen auf der ganzen Welt verteilt. Sie ist dabei in zahlreiche sog. „Fabrics“ aufgeteilt, die wiederum etliche Server beinhalten. Jeder Fabric hat einen „Fabric Controller“, ein System, welches die interne Verwaltung übernimmt [8].

Logisch ist Windows Azure für den Entwickler in zwei Bereiche unterteilt: den Compute-Service und den Storage-Service (Abb. 1). Die Datenbank gehört nicht direkt zu Windows Azure, sondern ist unter dem Namen „SQL Azure“ ausgelagert. Als Schnittstelle zu herkömmlichen IT-Systemen wird ein Dienst namens „AppFabric“ bereitgestellt [9].



**Abbildung 1.** Logische Sicht auf Windows Azure.

### 3.3 Compute-Service

Der Compute-Service ist der Dienst, der die Webanwendung hostet und ausführt. Er besteht aus mehreren Instanzen, die zwei Typen zuzuordnen sind. „Web Role“-Instanzen sind von außen über HTTP zugänglich; auf ihnen läuft der Teil des Systems, der die Schnittstelle zum Nutzer bildet. „Worker Role“-Instanzen sind am ehesten mit lokalen Serverprogrammen vergleichbar, sie funktionieren wie ein Hintergrundprozess.

Beispielsweise wäre ein Online-Videokonvertierer denkbar, bei dem ein Nutzer ein Video auswählen kann, Einstellungen treffen kann und dann auf „Konvertieren“ klickt. Nach ein paar Minuten kann er das konvertierte Video dann „abholen.“

Hier liefere die Webanwendung, in der der User die Einstellungen für eine Videokonvertierung trifft, in einer Web Role. Im Hintergrund läuft eine Worker Role, die dann Aufträge entgegennimmt und ein Video in der Cloud konvertiert, ohne dass der Nutzer in der Webanwendung bleiben muss.

Von diesen beiden Instanztypen kann es in einem Compute-Service (in der Windows-Azure-Verwaltungsoberfläche „Hosted Service“ genannt) beliebig viele Instanzen geben. Steigt die Last der Anwendung an, können einfach mehr Instanzen dazugeschaltet werden. Fällt sie wieder, kann die Zahl der Instanzen wieder reduziert werden.

Die Plattform skaliert also nicht automatisch. Jedoch existiert eine API, die es ermöglicht, ein automatisches Skalieren zu implementieren.

Zu beachten ist noch, dass die Web-Role-Instanzen zustandslos sind. So ist nicht garantiert, dass ein Nutzer während dem Nutzen einer Anwendung stets mit der gleichen Instanz arbeitet, und die Lastverteilung funktioniert besser. Um einen nutzerspezifischen Zustand zu erhalten, muss dieser in einem Storage Service, in einer Datenbank oder lokal gespeichert werden.

Um die Aktivität seiner Anwendungen zu beobachten, kann eine Logging-API gerufen werden. Jegliche Ausnahmebehandlung, z.B. wenn es mehrfach zu Abbrüchen kommt, muss jedoch selbst implementiert werden. Azure bietet nur die Möglichkeit, überhaupt auf diese Logging-Daten zuzugreifen.

Durch das Prinzip der Instanzen ist es dem Entwickler nicht möglich, direkt auf die virtuellen Maschinen zuzugreifen. Auch Anwendungen, die in einer Worker Role laufen, haben keinen administrativen Zugriff.<sup>2</sup> Der Vorteil dieser Vorgehensweise ist, dass man sich als Entwickler nicht um das Betriebssystem kümmern muss, beispielsweise indem man Updates installiert – dies geschieht alles automatisch auf Systemebene. So kann der administrative Aufwand minimal gehalten werden, während gewährleistet ist, dass die Anwendungen praktisch unterbrechungsfrei laufen können. Ein Entwickler kann sogar Updates seiner Anwendungen installieren, ohne einen Ausfall in Kauf zu nehmen [8].

### 3.4 Storage-Service

Der Storage-Service ist der Dienst, der die Speicherung von Daten jeder Art in der Cloud ermöglicht. Grundsätzlich bietet er drei Arten, Daten zu speichern: „Blobs“, „Tables“ und „Queues“.

Blobs<sup>3</sup> enthalten binäre Daten und sind in Containern organisiert. In ihnen können beispielsweise Bilder, Videos oder PDF-Dateien abgelegt werden.

Tables sind nicht, wie der Name vermuten lässt, relationale Tabellen; stattdessen sind es Konstrukte, die eher mit Bäumen vergleichbar sind. Folglich lassen sie sich auch nicht wie klassische Datenbanken mit SQL abfragen. Der Vorteil davon ist, dass diese Tables besser horizontal skalieren<sup>4</sup> als relationale Datenbanken. „[...] a single Windows Azure table can contain billions of entities holding terabytes of data.“<sup>5</sup> Queues sind die dritte Art, Daten mit dem Azure Storage Service zu speichern. Sie sind im Gegensatz zu Blobs und Tables Konstrukte, die der Kommunikation zwischen Web Role und Worker Role dienen. Im Beispiel des Online-Videokonvertierers würde die Web Role für jedes zu konvertierende Video einen Eintrag in eine Queue schreiben, die Worker Role kann dann diese Queue beobachten und alles abarbeiten, was in ihr steht.

Ein großer Vorteil des Storage-Service ist, dass man sich als Entwickler wiederum um nichts kümmern muss. Windows Azure selbst sorgt für Verfügbarkeit und Sicherheit der Daten.

Beim Anlegen eines Storage Service kann gewählt werden, in welcher Region die Daten abgelegt werden sollen. Müssen vertrauliche Daten beispielsweise von Rechts wegen innerhalb der EU bleiben, kann dies entsprechend konfiguriert werden. Sind die zu erwartenden Nutzer überwiegend Asiaten, kann

<sup>2</sup> „Elevated Privileges“ ist eine neue Funktion von Azure, die nun auch die Ausführung von Web- und Worker-Roles mit Administrationsrechten ermöglicht [10].

<sup>3</sup> Blob = Binary Large Object

<sup>4</sup> horizontales Skalieren = Skalieren durch Verteilen von Daten auf viele Server

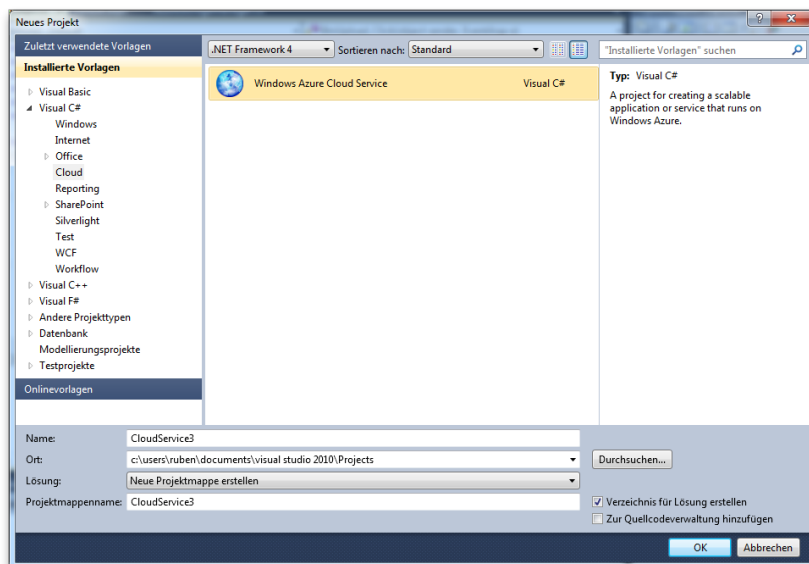
<sup>5</sup> [8], David Chappell: Introducing Windows Azure, S. 7

eingestellt werden, dass die Daten in Asien gelagert werden. Dabei legt Windows Azure alle Daten immer in zwei Serverfarmen in dieser Region ab, um ein ortsunabhängiges Backup zu haben.

Der Zugriff auf die per Storage-Service gespeicherten Daten muss nicht zwangsläufig über eine Azure-Anwendung erfolgen. Alles ist über Standard-URIs zu erreichen, also kann auf die Daten z.B. auch über eine Anwendung zugegriffen werden, die bei einem Webhoster oder in einer anderen Cloud liegen [8].

### 3.5 Datenbank

Die Datenbank ist, wie erwähnt, nicht direkter Bestandteil von Windows Azure, sondern als „SQL Azure“ ausgelagert. Sie kann wie eine „normale“ relationale Datenbank genutzt werden, indem man Anfragen per SQL<sup>6</sup> stellt – doch sie ist natürlich ebenfalls cloud-basiert [11].



**Abbildung 2.** Nach Installation des Windows Azure SDK bietet Visual Studio 2010 den „Windows Azure Cloud Service“ als Projekttyp an.

<sup>6</sup> SQL = Structured Query Language; Standardabfragesprache für relationale Datenbanken

RUBEN DEYHLE

## 4 Praxistest

Dieses Kapitel berichtet von der praktischen Arbeit mit Windows Azure, die größtenteils auf einem Lehrvideo von Microsoft im Rahmen des „Windows Azure Virtual Lab“ basiert [12].

Dabei wird beispielhaft eine einfache Webapplikation entwickelt und Schritt für Schritt „in die Cloud“ geschoben.

### 4.1 Voraussetzungen

Microsoft stellt ein Windows Azure Software Development Kit (SDK) für Visual Studio zur Verfügung, das primäre Tool zur Arbeit mit Windows Azure. Man ist aber nicht gezwungen, Windows und Visual Studio zu nutzen oder die Anwendung in ASP.NET zu schreiben. PHP, Ruby, Python und Java werden ebenfalls unterstützt, entsprechende SDKs und Integration in die IDE<sup>7</sup> Eclipse werden zum Download angeboten [13].

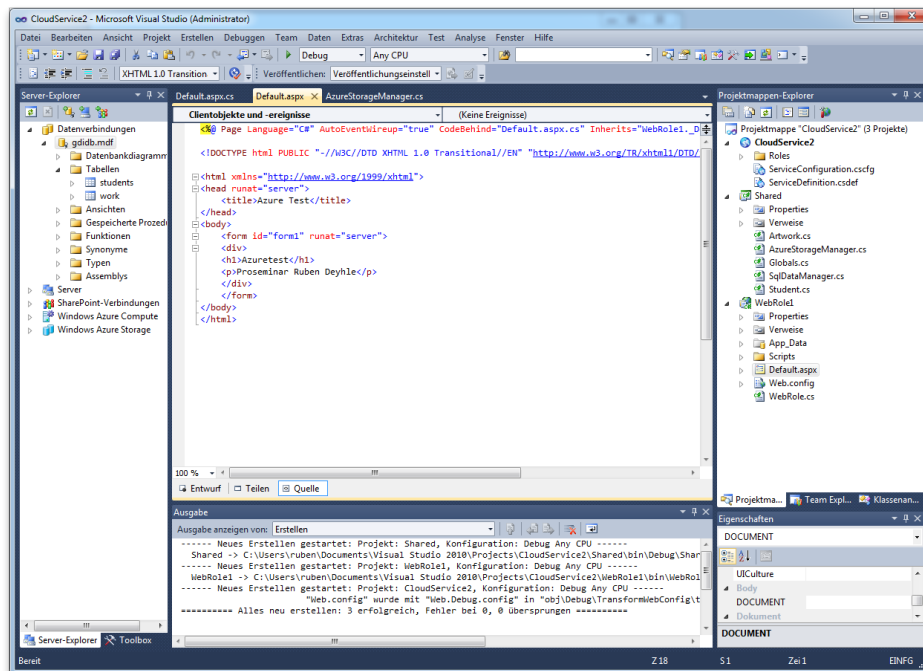


Abbildung 3. Windows-Azure-Entwicklung in Visual Studio 2010

<sup>7</sup> IDE = Integrierte Entwicklungsumgebung



Die Verwaltung von Windows Azure läuft über <http://windows.azure.com>, wo man sich mit einem Windows-Live-Account registrieren kann. Dafür werden Kreditkartendaten benötigt, jedoch wird diese natürlich nur belastet, wenn tatsächlich Kosten verursacht werden. Bis März 2011 wird ein „Introductory Special“ angeboten, das die grundlegende Funktionalität kostenlos anbietet [14].

## 4.2 Eine einfache Webanwendung

In Visual Studio besteht nach Installation der Azure Tools die Möglichkeit, einen „Windows Azure Cloud Service“ als neues Projekt anzulegen (Abb. 2). Für eine einfache Applikation wird darin eine „Web Role“ angelegt. Dies stellt die Basis für eine Webanwendung dar, die man nun in Visual Studio schreiben kann. Abbildung 3 zeigt Visual Studio 2010 bei der Entwicklung für Windows Azure. Noch ist dabei natürlich alles lokal gehostet.

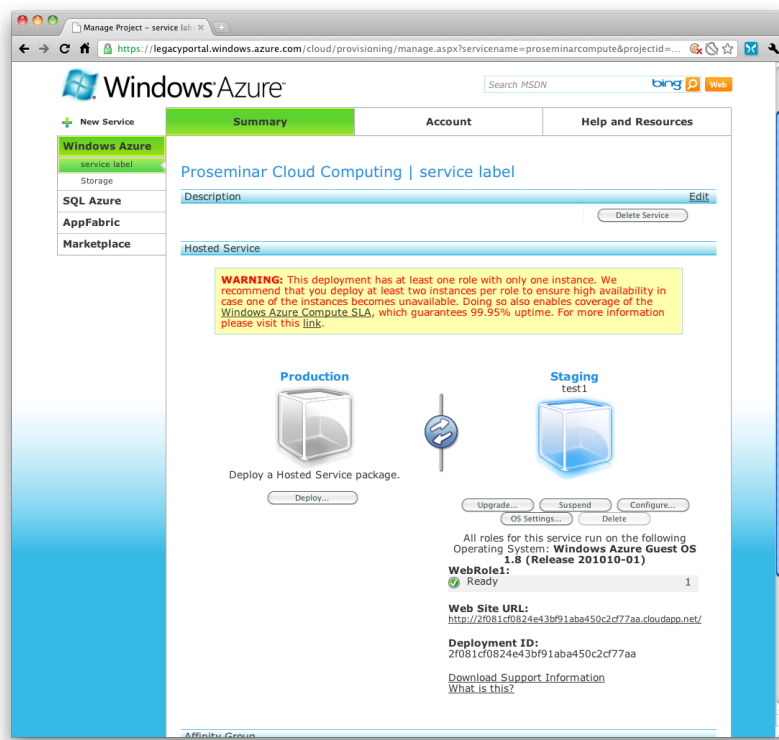
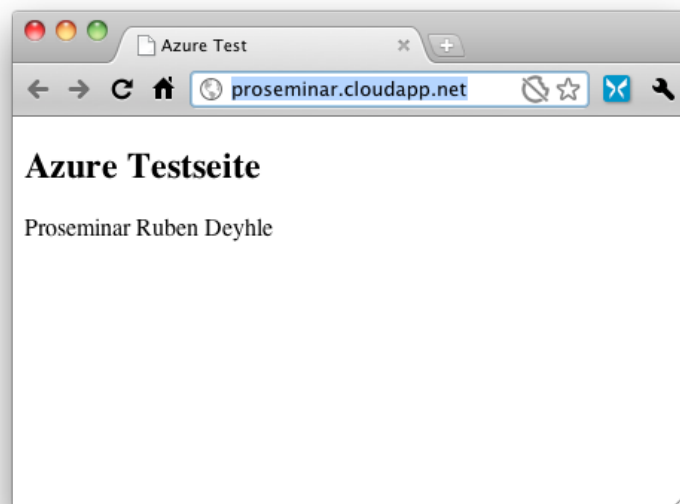


Abbildung 4. Staging- und Production-Bereich in der Verwaltungsoberfläche von Windows Azure.

Diese Anwendung kann dann direkt aus Visual Studio heraus auf Windows Azure geladen werden: nach Auswahl der „Publish“-Funktion öffnet sich der Webbrowser, der die Verwaltungsoberfläche von Windows Azure (<http://windows.azure.com>) aufruft (Abb. 4). Hier muss zuerst ein neuer Service gebucht werden. Zur Auswahl steht „Storage“ und „Hosted“, wobei „Hosted“ den Compute-Service von Windows Azure darstellt. Für eine Webanwendung wird also „Hosted“ benötigt. Anschließend wird direkt eine URL erstellt, über die die Webanwendung online erreichbar sein wird. Hier könnte auch schon gewählt werden, in welcher geografischen Region die Daten liegen sollen [12].

Innerhalb des „Hosted Service“ stehen zwei Bereiche, für produktiven Einsatz und für die Entwicklung („Staging“), bereit.<sup>8</sup> Das eigentliche Publishing besteht dann im Hochladen zweier Dateien, die Visual Studio erzeugt hat: ein Anwendungspaket und eine Konfigurationsdatei. Nach dem Hochladen kann die Webanwendung über die Verwaltungsoberfläche bequem gestartet werden und ist sodann über eine Subdomain von `cloudapp.net` erreichbar (Abb. 5) [12].



**Abbildung 5.** Mini-Webanwendung auf Windows-Azure-Servern

<sup>8</sup> Eine unter „Staging“ abgelegte Webanwendung ist dennoch online erreichbar, jedoch unter einer weniger „schönen“ URL.

### 4.3 Lokales Testen

Das Windows Azure SDK umfasst das „Windows Azure Simulation Environment“, das beim Start des Debug-Vorgangs in Visual Studio automatisch gestartet wird. Es simuliert lokal einen Windows Azure Fabric. So können Webanwendungen lokal getestet werden, als wären sie in der Cloud. Ein Problem tritt auf, wenn man externe Dienste wie den Windows-Live-Login-Service einbaut, damit man keine eigene Nutzerverwaltung programmieren muss. Wenn man diesen externe Dienst für die endgültige URL einrichtet, wird er beim lokalen Testen oder in der „Staging Area“ nicht funktionieren, da der Webanwendung dort eine andere Domain zugewiesen wird bzw. lokal nur über localhost erreichbar ist. Dies lässt sich lösen, indem man die entsprechenden DNS-Einträge lokal anpasst (unter Windows per hosts-Datei) [12].

### 4.4 Datenbank: SQL Azure

Auf der Verwaltungsoberfläche von Windows Azure kann unter „SQL Azure“ eine Datenbank angelegt werden. Neben maximaler Größe und einigen weiteren Einstellungen können auch Firewall-Einstellungen getroffen werden. Dies ist nützlich, um der eigenen IP-Adresse und somit dem eigenen Computer Zugriff auf die Datenbank zu gewähren, wenn noch nicht die komplette Anwendung in der Cloud läuft, sondern noch Teile lokal. Auf die Datenbank kann dann mit einem normalen Datenbankclient zugegriffen werden, beispielsweise Microsoft SQL Server Management Studio, um Tabellen anzulegen. Für den Zugriff aus der Webanwendung heraus müssen nur noch im Code die entsprechenden Zugangsdaten eingetragen werden, der Zugriff findet dann über Azure-Bibliotheken statt, die Bestandteil des SDK sind [12].

### 4.5 Datenspeicherung mit Storage

Um Daten in der Cloud abzulegen, gibt es den „Storage Service“, der genau wie ein „Hosted Service“ eine eindeutige URL beim Anlegen zugewiesen bekommen muss.

Wie auf die Datenbank kann dann über entsprechende Bibliotheken, die Bestandteil des Windows Azure SDK sind, auf den Storage Service zugegriffen werden.

Über die Verwaltungsoberfläche können noch diverse Einstellungen getroffen werden (Abb. 6), unter anderem kann die Nutzung eines CDN (Content Delivery Network) eingerichtet werden, um die Bandbreite zu optimieren [12].

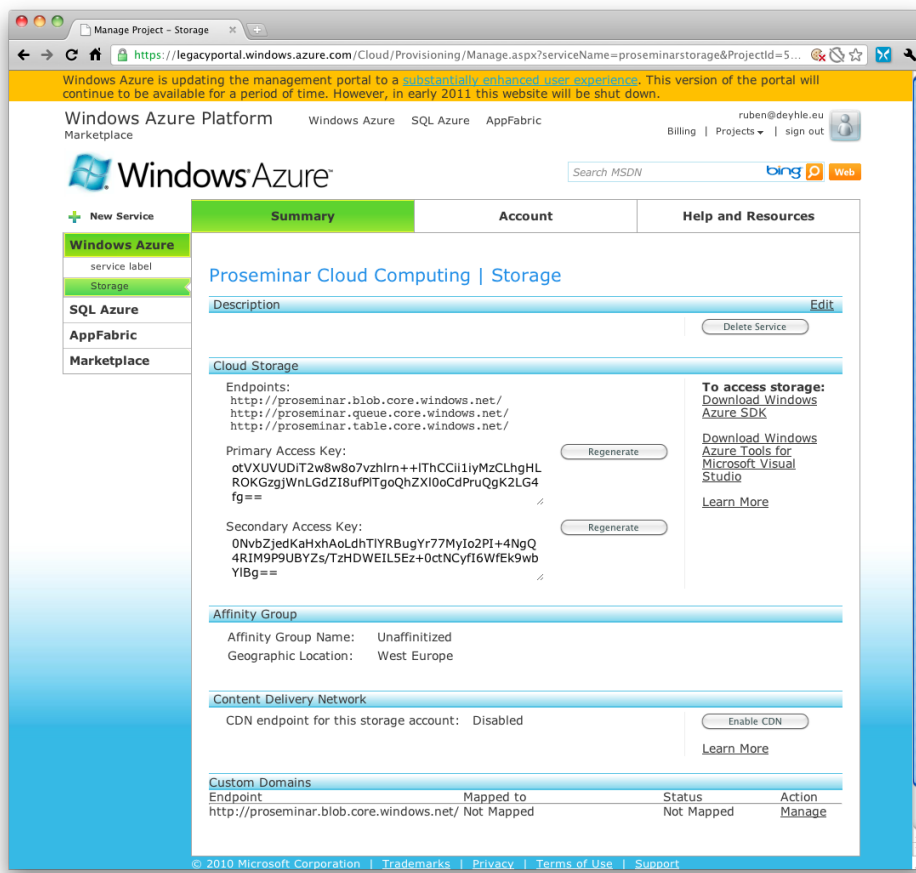


Abbildung 6. Storage-Bereich in der Verwaltungsoberfläche von Windows Azure.

## 5 Bewertung

Abschließend folgt nun eine kurze Bewertung von Windows Azure.

### 5.1 Vergleich von Windows Azure

**...mit Amazons Web Services:** AWS und Azure scheinen auf den ersten Blick recht ähnlich zu sein. Beide bieten Computing, Storage und eine Datenbank an; beide überlassen die Entwicklung von SaaS-Produkten dem Entwickler und stellen diesem dafür eine leistungsfähige Plattform zur Verfügung. Amazon geht dabei jedoch einen Schritt weiter als Microsoft und bietet mit der EC2<sup>9</sup> eine ganze Cloud-Infrastruktur an, die nicht nur das Ausführen von Software in der Cloud, sondern auch den Betrieb von virtuellen Maschinen und ganzen Netzwerken in der Cloud ermöglicht [5].

**...mit Salesforces Force.com:** Auch Force.com ist relativ ähnlich zu Azure, jedoch legt Salesforce mehr Wert auf fertige SaaS-Anwendungen, die vor allem Unternehmenskunden direkt einsetzen können, ohne zuerst selbst Aufwand in die Entwicklung von cloudbetriebender Software stecken zu müssen. Microsoft bietet zwar auch SaaS-Cloudanwendungen an, sieht diese aber als anderen Unternehmenszweig, getrennt von den Windows Azure Services für Entwickler an.

### 5.2 Fazit

„Cloud Computing“ ist definitiv nicht nur ein leeres Buzzword, sondern ein ernstzunehmender Trend der IT-Welt. Erfreulich ist, dass es bereits mehrere ähnliche Anbieter zur Auswahl gibt und sich der Aufwand, eine Anwendung für die Cloud zu schreiben, in Grenzen hält.

Der große Vorteil von Cloud Computing für Endanwender ist offensichtlich: die allgegenwärtige Verfügbarkeit eigener Daten. Für die Wissenschaft ist der Vorteil, dass auch sehr große Experimente schnell und günstig durchgeführt werden können. Und für die Industrie schließlich liegt der Vorteil darin, dass Systeme sehr schnell wachsen können. Wenn früher großer zeitlicher und finanzieller Aufwand erforderlich war, um eine Anwendung Millionen von Nutzern zur Verfügung zu stellen, so kann dies heute relativ schnell und günstig geschehen.

Windows Azure stellt dabei ein solides Konzept dar, an dem vor allem aus der Windows-Welt kommende Entwickler ihre Freude haben werden. Die Entwicklung von Anwendungen für Windows Azure funktioniert sehr ähnlich wie die klassische Anwendungsentwicklung in Visual Studio. Erfreulicherweise hat Microsoft seine PaaS-Plattform dabei dennoch relativ offen konzipiert,

<sup>9</sup> EC2 = Amazon Elastic Compute Cloud

und man ist nicht dazu gezwungen, Microsoft-Software zur Entwicklung zu nutzen.

Wer sich eher für einen schnellen, produktiven Einsatz von Cloud-Technologien in seinem Unternehmen interessiert, wird vermutlich bei Salesforce, Google oder den SaaS-Lösungen von Microsoft besser aufgehoben sein als bei Windows Azure. Auch wer gerne die volle Kontrolle über das zugrundeliegende System hat, wird mit Windows Azure nicht so recht glücklich werden und sollte sich lieber Amazons Cloud zuwenden. Aber wer sich nicht um low-level-Aufgaben kümmern möchte, sondern lieber eine zuverlässig gewartete und funktionierende Plattform nutzen möchte, um seine eigenen Anwendungen darauf zum Laufen zu bringen – und idealerweise schon mit Microsofts Entwicklungsumgebung vertraut ist – für den ist Windows Azure genau das Richtige.

## Literatur

1. Nikolai Zotow. Unter Dampf. Dienste aus dem Netz: Anbieter und Preise. iX Special 2/2010 – Cloud, Grid, Virtualisierung, S. 20-23
2. Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia. 2010. A view of cloud computing. Commun. ACM 53, 4 (April 2010), 50-58. DOI=10.1145/1721654.1721672 <http://doi.acm.org/10.1145/1721654.1721672>
3. Google App Engine Billing and Budgeting Resources, <http://code.google.com/intl/de-DE/appengine/docs/billing.html>, Stand: 05.12.2010
4. Daniel Hofmann. Lauschige Plätze. Dateien verteilen übers Web 2.0. iX Special 2/2010 – Cloud, Grid Virtualisierung, S. 24-32
5. Amazon AWS. <http://aws.amazon.com/de/>, Stand: 05.12.2010
6. Force.com. <http://www.salesforce.com/platform/>, Stand: 05.12.2010
7. Windows Azure Offers. <http://www.microsoft.com/windowsazure/offers/>, Stand: 07.12.2010
8. David Chappell. Introducing Windows Azure. Dezember 2009. <http://download.microsoft.com/documents/uk/mediumbusiness/products/cloudonlinesoftware/IntroducingWindowsAzure.pdf>
9. Windows Azure. <http://www.microsoft.com/windowsazure/>, Stand: 06.12.2010
10. Neue Features für Microsoft Cloud-Betriebssystem Windows Azure ab sofort verfügbar. <http://www.microsoft.com/germany/net/WindowsAzure/News.aspx?page=63>, Stand: 08.12.2010
11. SQL Azure. <http://www.microsoft.com/en-us/sqlazure/database.aspx>, Stand: 07.12.2010
12. Windows Azure Virtual Lab. <http://www.msdev.com/azure/vlab/>, Stand: 25.10.2010
13. Windows Azure Platform and Interoperability. <http://www.microsoft.com/windowsazure/interop/default.aspx>, Stand: 07.12.2010
14. Windows Azure Platform Introductory Special. <http://www.microsoft.com/windowsazure/offers/popup/popup.aspx?lang=en&locale=en-US&offer=MS-AZR-0001P>, Stand: 25.10.2010