

Sebastian Huppmann

## Automatische Erstellung einer SharePoint 2013 Entwicklungsumgebung in Microsoft Azure via PowerShell

Zeit und Geld sparen durch die Verlagerung in die Cloud  
(Erstveröffentlichung Juni 2014 in „VisualStudioOne“)

Microsofts Cloud Azure nimmt immer mehr Fahrt auf. Sie verbessert zunehmend nicht nur IT-Services in Unternehmen mit Produkten wie Office 365 und SharePoint-Online, sondern auch die Werkzeuge und Möglichkeiten für SharePoint-Entwickler. Das neue App-Modell für SharePoint sieht es vor Apps in der Cloud zu hosten. Das erhöht die Verfügbarkeit und die Wartbarkeit der Anwendungen. Die gleichen Vorteile entstehen auch beim Betrieb einer Entwicklungsumgebung für SharePoint 2013 in der Cloud. Wie einfach das geht und was für Vorteile es bringt, lesen Sie in den folgenden Zeilen.

Auf der diesjährigen SharePoint Conference in Las Vegas wurde überspitzt zu Beginn einer Session die Frage an SharePoint-Entwickler gestellt, ob sie denn immer noch ein kiloschweres Entwicklungsnotebook mit sich herumschleppen würden. Ca. 80% der Teilnehmer streckten auf diese Frage hin ihre Hände in die Höhe. Die Aussage des Speakers und Microsoft Mitarbeiters Richard diZerega darauf war: „You’re doing something wrong.“. Sie machten also etwas falsch. Es sei nicht mehr zeitgemäß lokal Arbeit an einem teuren Notebook zu verrichten und diverse virtuelle Maschinen für SharePoint darauf zu hosten, so der Vortragende. Anstatt dessen müssten die Vorteile von Microsoft Azure genutzt werden.

### Kalkulierbarkeit einer Entwicklungsumgebung

Mit Hilfe von Windows Azure können die Kosten für eine Entwicklungsumgebung für jedes Projekt prognostiziert und kalkuliert werden. Der Azure Pricing Calculator ermöglicht es die Betriebsstunden der benötigten virtuellen Maschinen (VMs) auf die Stunde bezogen auszurechnen. Zusätzlich gibt er Auskunft wie teuer der physische Speicher in der Cloud für die jeweilige Umgebung ist. (Azure Pricing Calculator: <http://www.windowsazure.com/en-us/pricing/calculator>)

In Abbildung 1 ist eine Kalkulation zur einer trivialen SharePoint 2013 Entwicklungsumgebung, die an 20 Tagen im Monat zu je acht Stunden läuft zu erkennen. (Die praktische Erstellung genau dieser Umgebung findet sich ab dem Abschnitt *Praktische Einrichtung der Entwicklungsumgebung*):

Art	Stück	Preis	Kosten
Active Directory Medium VM (2 x 1.6GHz CPU, 3.5GB RAM)	160 <u>hr</u>	0.135 € / <u>hr</u>	21.6 €
SP 2013 Extra Large VM (8 x 1.6GHz CPU, 14GB RAM)	160 <u>hr</u>	0.269 € / <u>hr</u>	43.04 €
Storage (25 GB + 150 GB) <u>Geo Redundant</u>	175 GB	12.39 € / <u>mo</u>	12.39 €
<b>Gesamt</b>			<b>77.03 €</b>

Abbildung 1: Kalkulation einer SharePoint 2013 Entwicklungsumgebung in Microsoft Azure

Die entstehenden Kosten können somit direkt einem Projekt zugeordnet werden und der jeweiligen Kostenstelle bzw. dem Kunden in Rechnung gestellt werden. Anhand der Gesamtsumme von 77,03 € im Monat für eine triviale SharePoint-Farm, beispielsweise zum Entwickeln einer SharePoint-App, kann man das Einsparungspotenzial gegenüber teuren Entwicklernotebooks leicht errechnen.

## **Die Vorteile einer Entwicklungsumgebung in der Cloud im Überblick**

*Eine Entwicklungsumgebung in der Cloud führt zu einer besseren Kollaboration innerhalb des Entwicklungsteams.* Eine lokale Installation einer Entwicklungsumgebung auf dem Notebook eines Teammitglieds ist für andere Teammitglieder nur schwer erreichbar und kann nur schlecht geteilt werden. Durch die Verlagerung in die Cloud ist die aktuelle Umgebung stets verfügbar und unabhängig von einem einzelnen Notebook. Eine Remote-Desktop-Verbindung für ein neues Teammitglied zum Server innerhalb von Microsoft Azure ist binnen Sekunden erstellt.

*Durch die Möglichkeit die physischen Daten geo-redundant zu Speichern erhöht sich die Ausfallsicherheit.* Microsoft Azure bietet an die virtuellen Festplatten eines Servers nicht nur lokal redundant zu, sondern auch geo-redundant zu speichern. Das heißt, dass die Daten nicht nur innerhalb eines Rechenzentrums von Azure gespeichert werden können, sondern auch parallel in einem weiteren Rechenzentrum auf einem anderen Kontinent. Das erhöht die Ausfallsicherheit der Entwicklungsumgebung.

*Durch die flexible Skalierbarkeit von Microsoft Azure ist eine Anpassung der Entwicklungsumgebung an die momentane Projektsituation besser möglich.* Wenn weitere Server für eine SharePoint-Farm innerhalb eines Projekts benötigt werden, können diese einfach der Farmkonfiguration hinzugefügt werden. Notebook-Umgebungen stoßen hier ressourcentechnisch schnell an ihre Grenzen. Ebenso lassen sich ganz einfach einem existierendem Microsoft Azure-Server mehr Rechenleistung und Arbeitsspeicher hinzufügen. Auch zusätzliche virtuelle Festplatten sind schnell erstellt und einem Server zugewiesen.

*Die Cloud ermöglicht eine leichtere Bereitstellung von temporären Show-Case und Testsystemen für den Kunden.* Gerade im Bereich SharePoint wo es sehr viele Drittanbieter mit herausragenden Lösungen gibt, die Ergänzungen zum eigentlichen Produkt anbieten, wird der Wunsch nach einem temporären Test- oder Show-Case-System oft schnell laut. Viele Kunden wünschen sich nur für ein paar Tage beispielsweise eine Nintex-Workflow-2013-Testumgebung oder eine K2-Testumgebung um ihre Workflow-Anforderungen an die Software evaluieren zu können. Mit der Erstellung eines Images aufbauend auf einer oder mehreren virtuellen Festplatten kann per Knopfdruck ein solches System bereitgestellt werden und nach Gebrauch wieder abgeschaltet werden.

*Leicht zugängliche Q- und Staging-Systeme bieten für den Kunden mehr Transparenz und Einsicht in das Projekt.* Es ist möglich Q- und Staging-Systeme für den Kunden einzurichten, mit deren Hilfe in Zusammenarbeit unter agilen Entwicklungsmethoden die nächsten Schritte des Projekts am derzeitigen Entwicklungsstand geplant, präsentiert und besprochen werden können. Durch ein Q-System in der Cloud kann dem Kunden stets der aktuelle Entwicklungsstand gezeigt werden. Er erhält somit ein sichereres Gefühl, weil er den Fortschritt des Projekts sieht und Einfluss nehmen kann auf eventuelle Abweichungen.

*Microsoft ermöglicht eine bessere Homogenität der Entwicklungsumgebungen durch die Bereitstellung von Images, die nach den Best Practices für SharePoint 2013 erstellt wurden.* Man erhält somit reinere und homogene Entwicklungsumgebungen, die nicht unterschiedliche Farmkonfigurationen aufweisen oder von mehreren älteren Projekten „verseucht“ sind. Wizard-Installationen von SharePoint oder beispielsweise unterschiedliche Varianten von AutoSPInstaller-Skripten, die während eines Entwicklungsprozesses unerwünschte Verhaltensweisen der Anwendung hervorrufen, werden somit vermieden.

## **Praktische Einrichtung der Entwicklungsumgebung**

Im Folgenden soll eine SharePoint 2013 Entwicklungsumgebung in Microsoft Azure bereitgestellt werden. Abbildung 2 verdeutlicht die Bestandteile der SharePoint-Farm. Die drei Hauptkomponenten lauten wie folgt:

- Active Directory (AD) - Medium VM - Windows Server 2012 R2 (2 x 1.6GHz CPU, 3.5GB RAM)
- SP 2013 Farm inkl. Visual Studio 2013 und SQL Server 2012 - Extra Large VM - Windows Server 2012 R2 (8 x 1.6GHz CPU, 14GB RAM)
- Storage Geo Redundant (25 GB + 150 GB)

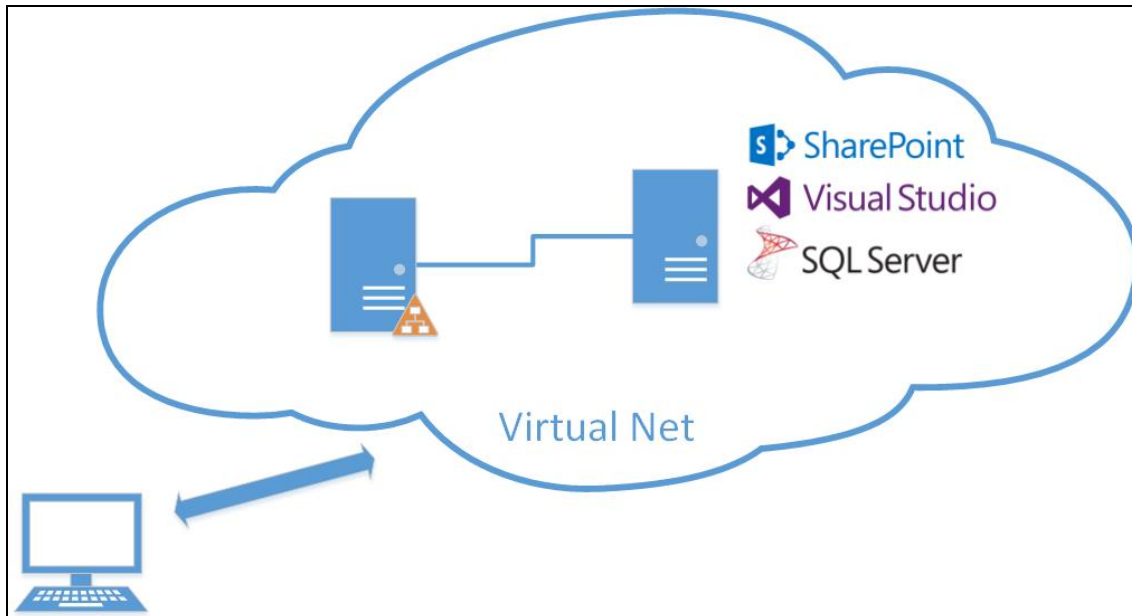


Abbildung 2: Big Picture der SharePoint 2013 Entwicklungsumgebung in Microsoft Azure

## Grundvoraussetzungen

Es muss ein eigener MSDN-Premium-Account oder Co-Administrator-Zugriff auf einen solchen vorliegen um auf die entsprechenden Microsoft-Images zugreifen zu können.

Um zu erfahren wie man einen Co-Administrator einrichtet, bitte diesen Link zu folgen:

<http://msdn.microsoft.com/en-us/library/windowsazure/gg456328.aspx>

## Schritt 1: Windows Azure PowerShell

Die spätere Installation der VMs in Microsoft Azure findet ausschließlich mit PowerShell statt. Dazu ist es notwendig die cmdlets für Microsoft Azure zu downloaden. Dies findet in der Regel über den Microsoft Web Platform Installer statt (siehe Abbildung 3).

Eine ausführliche Anleitung findet sich hier:

<http://www.windowsazure.com/en-us/documentation/articles/install-configure-PowerShell/#Install>

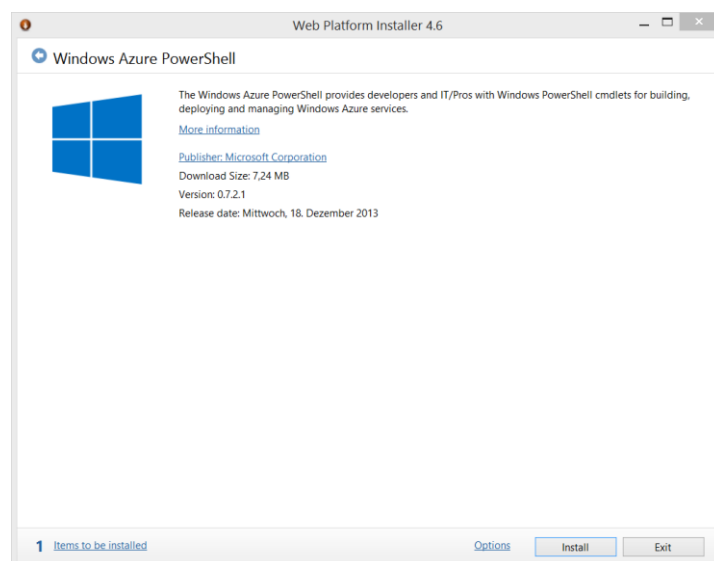


Abbildung 3: Web Platform Installer 4.6 für Windows Azure PowerShell

## Schritt 2: Management Zertifikat für Microsoft Azure

Um alle cmdlets für PowerShell ausführen zu dürfen muss man von seiner aktuellen Maschine aus ein sogenanntes Management-Zertifikat erstellen und es in das Microsoft Azure Portal hochladen (siehe Abbildung 4).

```
makecert -sky exchange -r -n "CN=<CertificateName>" -pe -a sha1 len 2048 -ss My
"<CertificateName>.cer"
```

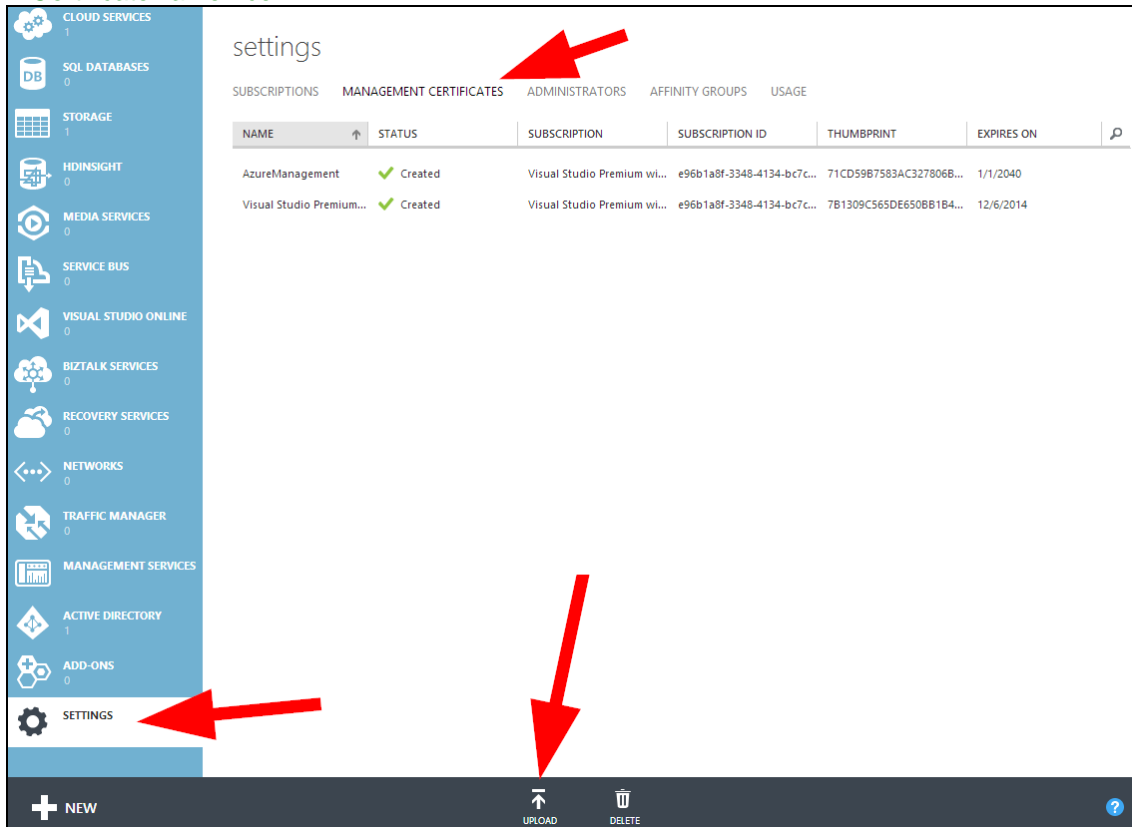


Abbildung 4: Azure Portal – Management Certificates

Eine ausführliche Anleitung findet sich hier: <http://msdn.microsoft.com/en-us/library/windowsazure/gg551722.aspx>

## Schritt 3: Einrichten der PowerShell ISE

Für ein angenehmeres Arbeiten empfiehlt es sich die PowerShell ISE einzurichten.

Um die cmdlets für Azure zu verwenden zu können, muss stets das entsprechende Modul importiert werden bevor man auf sie zugreift:

```
Import-Module "C:\Program Files (x86)\Microsoft SDKs\Windows Azure\PowerShell\Azure\Azure.psd1"
```

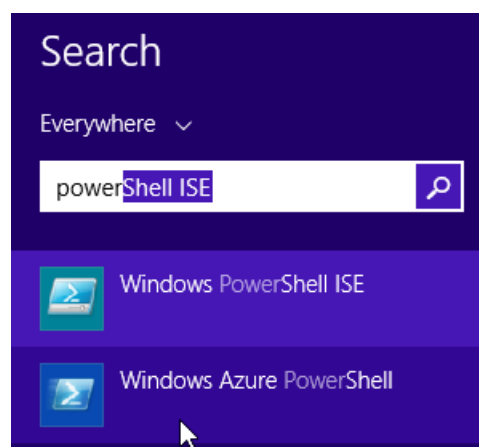


Abbildung 5: PowerShell ISE in der Windows 8.1 Search

## Schritt 4: Download der Installation-Skripte

Die Installation-Skripte werden von Microsoft bereitgestellt und sind hier zu finden:

<http://gallery.technet.microsoft.com/scriptcenter/Visual-Studio-2013-Gallery-60724f8b>

Die Installationsprozedur besteht im Wesentlichen aus drei Teilen, die separat nacheinander gestartet werden müssen.

## Schritt 5: Anlegen des Virtuellen Netzwerks

Zunächst wird mit dem Ausführen von *CreateVNet.ps1* ein virtuelles Netzwerk erstellt in welchem später die zwei virtuellen Maschinen laufen werden. Wie bereits erwähnt eine mittelgroße Maschine mit Active Directory (AD) und eine größere Maschine mit einer SharePoint 2013 Installation, einem SQL Server 2012 und entsprechenden Entwicklungstools wie dem Visual Studio 2013.

```
CreateVNet.ps1 -azurePublishSettingsFile "MSDNcredentials.publishsettings" -  
subscriptionName "Visual Studio Premium with MSDN" -vnetConfigurationPath "NetConfig.xml" -  
-affinityGroupName "AffinityGroup1" -affinityGroupLocation "North Europe"
```

Hierbei sind folgende Parameter speziell zu beachten:

- vnetConfigurationPath: In der referenzierten XML-Datei kann der Name des gewünschten Netzwerks und die IP-Adressen Konfiguration angepasst werden.
- azurePublishSettingsFile: Die publishsettings-Datei bekommt man auf einem der folgenden Wege. Entweder man ruft diesen Link auf: <https://windows.azure.com/download/publishprofile.aspx> oder man gibt folgendes cmdlet ein „*Get-AzurePublishSettingsFile*“. Bitte beachten Sie dazu den folgenden Sicherheitshinweis.

### Sicherheitshinweis

Die publishsettings-Datei enthält die Anmeldedaten (unverschlüsselt), die verwendet werden, um Ihre Microsoft Azure-Abonnements und Dienstleistungen zu verwalten. Die bewährte Sicherheitsmethode für diese Datei ist es, sie vorübergehend zu speichern und außerhalb Ihrer Quellverzeichnisse (zum Beispiel im Ordner Libraries\Documents). Löschen Sie sie, nachdem die Einstellungen importiert wurden. Ein böswilliger Benutzer könnte mit dem Zugang zur publishsettings-Datei Ihre Microsoft Azure-Dienste bearbeiten oder sie gar löschen.

## Schritt 6: Anlegen des AD-Servers

Im Anschluss muss das Skript *CreateNewADDSForest.ps1* ausgeführt werden. Hier wird die kleinere VM mit dem Active Directory erstellt und gestartet/aktiviert.

```
CreateNewADDSForest.ps1 -azurePublishSettingsFile "MSDNcredentials.publishsettings" -  
subscriptionName "Visual Studio Premium with MSDN" -storageAccountName  
"storageAccName" -vmName "SharePointAD" -vmSize "Medium" -adminUserName "SPAdmin"  
-adminUserPassword "pass@word1" -domainName "sharepoint.domain.com" -  
safeModeAdminPassword "pass@word1" -domainSPFarmAccountName "SPFarm" -  
domainSPFarmAccountPassword "pass@word1" -vnetName "SharePointNet" -subnetNames  
"Subnet1"
```

Hierbei werden wiederum einige wichtige Parameter verlangt. Sie lauten wie folgt:

- storageAccountName: Der Storage Account unter dem später die VMs gespeichert werden. Man muss Wert darauf legen, ob die VMs geo-redundant oder nur lokal redundant gespeichert werden sollen. Das wirkt sich auf den Endpreis und die Ausfallsicherheit aus und kann im Azure Portal eingestellt werden, auch zu einem späteren Zeitpunkt.
- vmName: Der Name der VM für das Active Directory, z.B. „SharePointAD“

- vmSize: Die Größe der VM. „Medium“ wird für ADs empfohlen.
- adminUserName: Admin-Account für die VM
- adminUserPassword: Passwort des Admin-Accounts der VM
- domainName: gewünschter Domainname, z.B.: sharepoint.domain.com
- safeModeAdminPassword: Das Passwort für den safe Mode des Admin-Accounts der VM
- domainSPFarmAccountName: Der administrative Farm-Account zum Betrieb
- domainSPFarmAccountPassword: Passwort für den administrativen Farm-Account
- vnetName: Name des vorher angelegten virtuellen Netzwerks unter Schritt 6
- subnetNames: Name des vorher angelegten virtuellen Subnetzes unter Schritt 6

Da dieser Schritt einige Minuten in Anspruch nehmen kann, wurde er mit einer Statusleiste versehen (siehe Abbildung 6).

```

Install-ADDSForest.
.

Validating environment and user input.
All tests completed successfully.

Installing new forest.
Creating directory partition: CN=Configuration,DC=customer,DC=alegri,DC=eu: 652 objects remaining.

PS C:\WINDOWS\system32> C:\Users\shuppmann\Desktop\WindowsAzure.ps1
PS C:\WINDOWS\system32> C:\Users\shuppmann\Desktop\WindowsAzure.ps1
02/11/2014 15:09:26: Start to create virtual machine: CustomerNet.

OperationDescription                                OperationId                                           OperationStatus
-----
New-AzureVM                                          4ae07bd6-07a0-69f4-b796-68dfa2c85a6                Succeeded
New-AzureVM                                          25d5ebbd-bb59-6b30-bddd-385ab500820c                Succeeded
02/11/2014 15:14:36: Start to download and install the remoting cert (self-signed) to local machine trusted root.
02/11/2014 15:14:52: Start to configure Active Directory Forest via powershell remoting uri: https://customer.net.cloudapp.net:61532/.
(WM) 02/11/2014 14:15:31: Start to install AD-Domain-Service feature.
(WM) 02/11/2014 14:18:18: Start to install ADDSForest.
  
```

Abbildung 6: Anlegen des Active Directory Server in Microsoft Azure via PowerShell

## Schritt 7: Anlegen des Entwicklungsservers inkl. SharePoint 2013 Installation

Zuletzt muss das Skript *CreateSharePointDeveloperMachineInDomain.ps1* ausgeführt werden. Hier wird die größere VM inkl. Entwicklungsumgebung erstellt und gestartet/aktiviert.

```

CreateSharePointDeveloperMachineInDomain.ps1 -imageName
"03f55de797f546a1b29d1b8d66be687a__Visual-Studio-2013-Ultimate-12.0.21005.1-Update1-
AzureSDK-2.2" -azurePublishSettingsFile "MSDNcredentials.publishsettings" -
subscriptionName "Visual Studio Premium with MSDN" -storageAccountName
"storageAccName" -vmName "SharePointDev" -vmSize "ExtraLarge" -adminUserName
"SPAdmin" -adminUserPassword "pass@word1" -domainDnsName "sharepoint.domain.com" -
domainName "sharepoint" -domainUserName "SPAdmin" -domainUserPassword
"pass@word1" -domainSPFarmAccountName "SPFarm" -domainSPFarmAccountPassword
"pass@word1" -vnetName "SharePointNet" -subnetNames "Subnet1"
  
```

Die wichtigsten Parameter hierbei lauten:

- imageName: Dieses kann sich im Laufe der Zeit ändern. Die Images werden ständig aktualisiert. Es empfiehlt sich hin und wieder das folgende cmdlet zu starten und zu prüfen ob das Image noch aktuell ist: *Get-AzureVMImage | Select ImageName*
- vmName: Der Name der VM für die SP 2013 Entwicklungsmaschine, z.B. „CustomerSPDev“
- vmSize: Die Größe der VM. „ExtraLarge“ wird hier empfohlen.

Die Ausführung dieses Skripts nimmt am meisten Zeit in Anspruch, da zum einen eine Instanz des SQL Servers 2012 aktiviert wird und zum anderen der SharePoint 2013 Server installiert werden muss. Die Konsole liefert dabei stets den aktuellen Zwischenstand (siehe Abbildung 7), so kann mitverfolgt werden, wann beispielsweise die Central Administration angelegt wird.

```
(VM) 02/11/2014 14:54:02: SharePoint Farm is now running.
(VM) 02/11/2014 14:54:02: Start to create the Central Administration site on port 11111.
(VM) 02/11/2014 14:55:52: Start to install help collections.
(VM) 02/11/2014 14:55:52: Start to initialize security.
(VM) 02/11/2014 14:55:56: Start to install services.
(VM) 02/11/2014 14:56:14: Start to register features.
(VM) 02/11/2014 14:57:28: Start to install application content.
(VM) 02/11/2014 14:57:33: Start to create web application.
```

Abbildung 7: Installation der SharePoint 2013 Farm

## Schritt 8: Verbinden zum Entwicklungssystem via Remote Desktop Protocol

Sobald die Maschine erstellt ist kann sich damit verbunden werden. Mit einem Klick auf auf Connect wird die passende .rdp-Datei heruntergeladen. Zum Verbinden müssen die Credentials aus dem jeweiligen Skript (s.o.) verwendet werden.

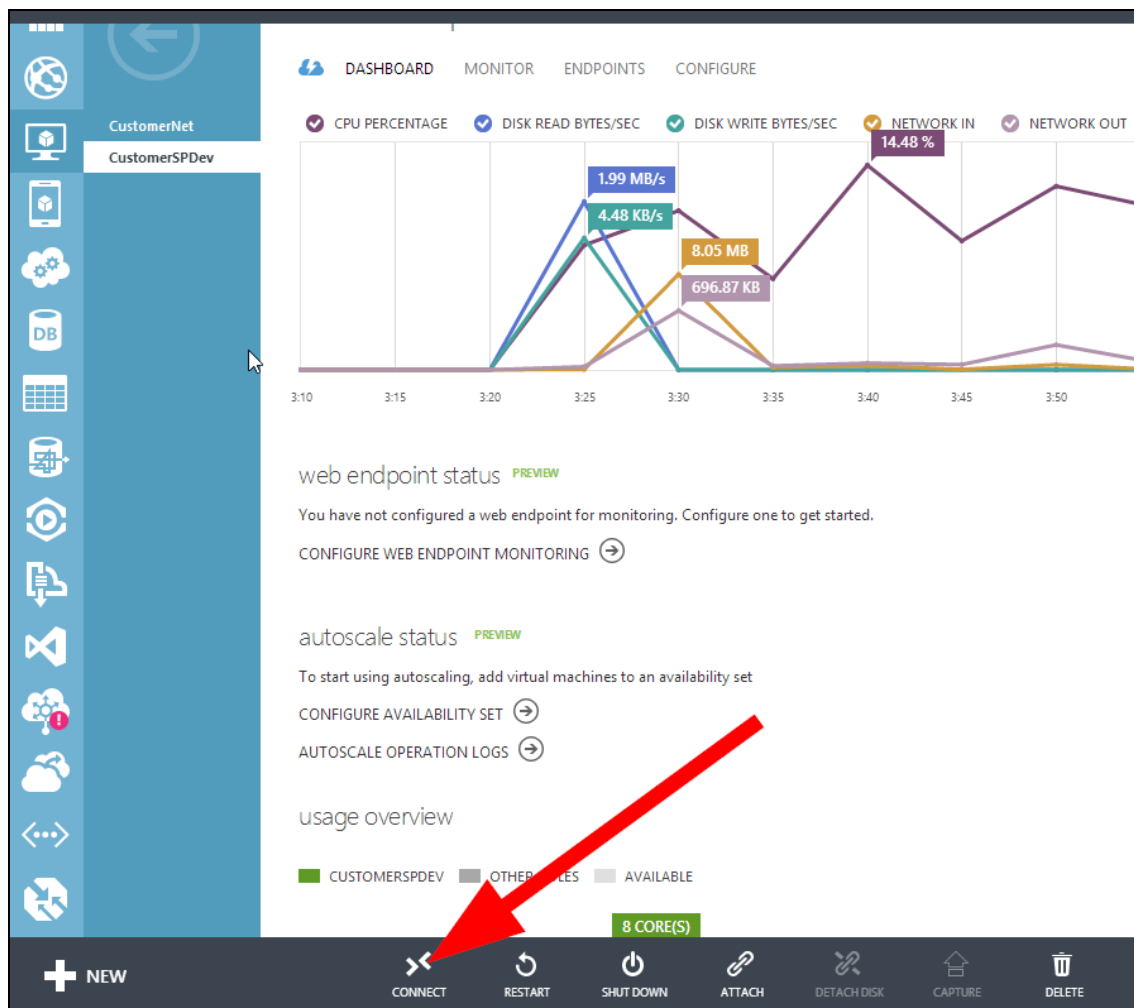
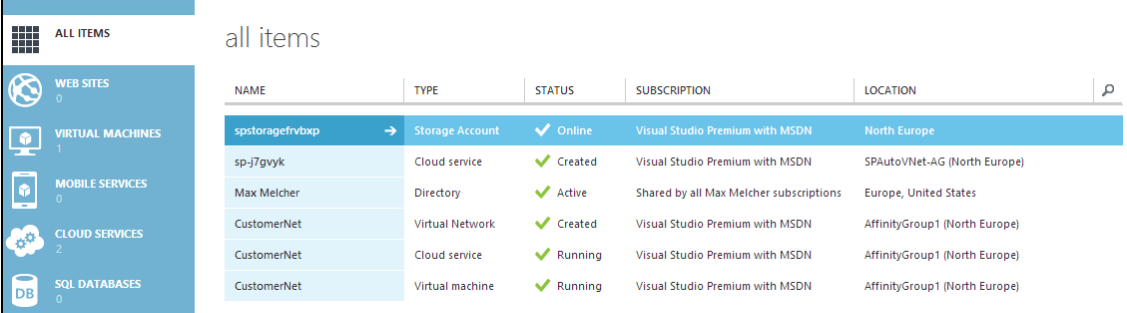


Abbildung 8: Azure Portal – Connect klicken für den Download einer .rdp-Datei

## Prävention unnötiger Laufzeitkosten für Azure

Um unnötige Laufzeitkosten zu vermeiden empfiehlt es sich das Azure Portal regelmäßig zu kontrollieren. Im Speziellen in der „all items“-Ansicht, da hier gut erkannt werden kann, welche Instanzen bzw. Services alle im Status „Running“ sind.



NAME	TYPE	STATUS	SUBSCRIPTION	LOCATION
spstoragefrvbxp	Storage Account	✓ Online	Visual Studio Premium with MSDN	North Europe
sp-j7gvyk	Cloud service	✓ Created	Visual Studio Premium with MSDN	SPAutoVNet-AG (North Europe)
Max Melcher	Directory	✓ Active	Shared by all Max Melcher subscriptions	Europe, United States
CustomerNet	Virtual Network	✓ Created	Visual Studio Premium with MSDN	AffinityGroup1 (North Europe)
CustomerNet	Cloud service	✓ Running	Visual Studio Premium with MSDN	AffinityGroup1 (North Europe)
CustomerNet	Virtual machine	✓ Running	Visual Studio Premium with MSDN	AffinityGroup1 (North Europe)

Abbildung 9: Azure Portal – Übersicht über alle Einheiten

Ebenso sollte oft auf die aktuelle Rechnung geschaut werden (siehe „View my bill“ in Abbildung 9):

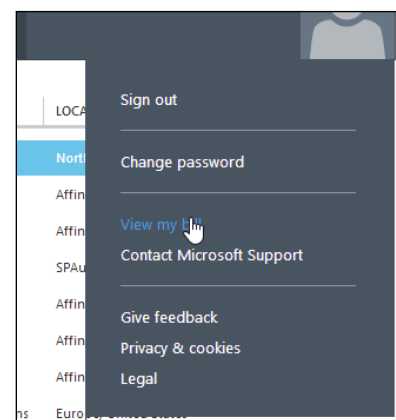


Abbildung 10: Azure Portal – View my Bill

Bei Erreichung des Limits können Azure VMs automatisch heruntergefahren werden. Beim MSDN-Account Premium ist dies zum Beispiel der Fall.

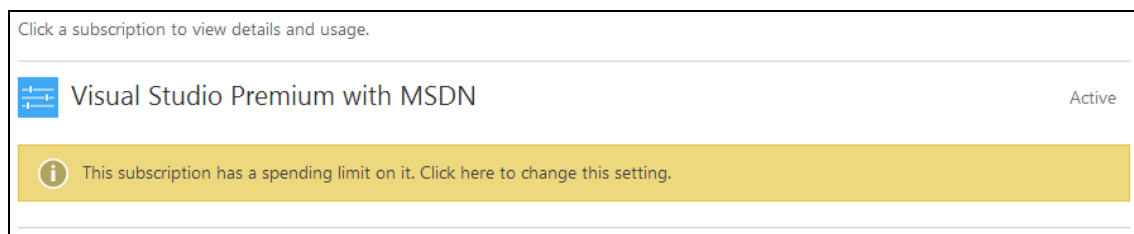


Abbildung 11: Azure Portal – Hinweis für Limitierung des Budgets

## Ausblick

Aus Sicht eines IT-Beraters sind die Hostingmöglichkeiten von Microsoft Azure eine wahre Erleichterung für das tägliche Business. Das Teilen und Präsentieren von neuen Lösungsansätzen und Projektfortschritten fällt erheblich leichter, weil ein Zugang zur Cloud durch das Internet in der Regel immer vorhanden ist. Die Erstellung von neuen Entwicklungsumgebungen und das einfache Kreieren von Templates für die unterschiedlichsten Szenarien sind technisch nicht nur sehr schnell realisierbar, sondern funktioniert stets zuverlässig. Es ist somit nur noch eine Frage der Zeit, dass sich unsere Entwicklungsumgebungen komplett in die Cloud verlagern und die Entwickler sich endgültig ihres schweren Notebooks entledigen können um mit leichteren Gerät remote zu arbeiten.



## Links und Quellen

<http://msdn.microsoft.com/en-us/library/windowsazure/gg551722.aspx>

<http://www.windowsazure.com/en-us/documentation/articles/install-configure-powershell/#Install>

<http://msdn.microsoft.com/en-us/library/windowsazure/gg456328.aspx>

<http://www.kerrb.com/>

[http://visualstudio2013galleryimage.azurewebsites.net/#\\_Toc369786641](http://visualstudio2013galleryimage.azurewebsites.net/#_Toc369786641)

<http://www.windowsazure.com/en-us/community/blog/>

<https://twitter.com/AzureNews>

<https://twitter.com/WindowsAzure>

<http://weblogs.asp.net/scottgu/archive/2013/12/12/windows-azure-new-scheduler-service-read-access-geo-redundant-storage-and-monitoring-updates.aspx>

<http://blogs.msdn.com/b/avkashchauhan/archive/2012/05/10/downloading-windows-azure-publish-settings-subscription-configuration-file.aspx>

## Autorenbox

Sebastian Huppmann ist Software Architekt und Senior Consultant bei dem in München sesshaften IT-Beratungsunternehmen Alegri International Service GmbH. Seit seinem Informatikstudium beschäftigt er sich mit cloud-basierten Architekturen und der maßgeschneiderten Entwicklung von Webanwendungen mit Schwerpunkt auf Microsoft Technologien. Sebastian leitet Implementierungen für die TOP-30 DAX Unternehmen. Auf Konferenzen und als Autor vermittelt er sein Wissen zu Lösungen rund um SharePoint und Microsoft Azure.

