Access Control Systems Lab
Sommersemester 2017

Praktikumsbericht 4

Jean-Marc Hendrikse

Prof. Dr. Hannes Hartenstein, Alexander Degitz, Jan Grashöfer, Till Neudecker
Forschungsgruppe Dezentrale Systeme und Netzdienste
Karlsruher Institut für Technologie (KIT)

Einleitung

In diesem Bericht stellen wir den Umgang mit *Microsoft Azure* vor. Microsoft Azure (oder auch nur *Azure*) ist eine Cloud-Computing-Plattform der Firma Microsoft [1]. Es bietet Dienste wie *SQL Azure* oder *AppFabric* an. Hauptsächlich wird bei Azure auf *Infrastructure as a Service* (IaaS), *Platform as a Service* (PaaS) und *Software as a Service* (SaaS) gesetzt [2].
Nachdem wir ein neues Microsoft-Konto über `https://account.microsoft.com/account` erstellt haben und ein konstenloses Azure-Konto angelegt haben, können wir damit beginnen uns tiefer mit Azure zu befassen und uns mit dem Umgang besser vertraut machen. Dafür erstellen wir zunächst ein *Windows Server 2012 R2 Datacenter* im Azure Portal und installieren *Microsoft Remote Desktop*, falls dieser nicht bereits vorher auf dem System vorhanden ist. Bei der Erstellung des Windows Server wird eine `rdp`-Dtaei (in unserem Fall `DC.rdp`) generiert, die wir über Remote Desktop ausführen können. Dadurch erhalten wir Zugriff auf unseren Windows Server 2012 und können auf einem blanken System konfigurationen und Installationen durchführen. Diese blanke Version des Windows Servers dient uns als Basuis für weitere Konfigurationen. So wollen wir zum einen in Kapitel 1 den Domain Controller näher vorstellen, der das Herzstück der Active Directory Domain Services bildet, und in Kapitel 2 installieren wir zur Verzeichnissynchronisation *Azure Active Directory Connect*.

1 Domain Controller

Seit der Windows Server Version 2008 ist der klassiche Active Directory (AD) in Rollen unterteilt. Die Kernkomponente bildet die Server-Rolle *Active Directory Domain Services* (**AD DS**). Mithilfe des AD DS können Administartoren Informationen zu Ressourcen eines Netzwerkes ganz einfach verwalten und in verteilten Datenbanken abspeichern [3]. Ebenso ist das Erstellen von Hierarchien für Benutzer und Computer eines Netzwerks möglich. In der Remote Desktop Konsole installieren wir auf dem Windows Server 2012 über das Tool *Server Manager* nun genau diesen Dienst.
Im ersten Schritt legen wir drei Benutzer, zwei Gruppen und zwei Zugriffsberechtigungen

[1] `https://www.microsoft.com`
[2] `https://de.wikipedia.org/wiki/Microsoft_Azure`
[3] `http://searchwindowsserver.techtarget.com/definition/Microsoft-Active-Directory-Domain-Services-AD-DS`

(engl. *permissions*) auf eine Datei an. Da wir noch keiner Domäne angehören, machen wir das nicht über das AD, sondern stattdessen lokal über das *Computer Management*, das wir über *Tools* > *Computer Management* erreichen und dort User und Gruppen verwalten. Wir legen zunächst mit einem Standardpasswort versehen drei lokale Benutzer an: *Test1*, *Test2* und *Test3*. Außerdem legen wir die beiden Gruppen *Testgroup12* und *Testgroup13* an, denen wir jeweils Benutzer zuordnen. In der Gruppe *Testgroup12* tragen wir die Benutzer *Test1* und *Test2* ein und in der Gruppe *Testgroup13* werden die beiden Benutzer *Test1* und *Test3* hinzugefügt. Jetzt haben wir Dummy-Benutzer und -Gruppen, mit denen man Zugriffsentscheidungen sehr gut simulieren kann. In dem *Dokumenten*-Ordner des angemeldeten Admin-Benutzers legen wir eine Datei namens `access_test.txt` an. Über Rechtsklick gelangen wir über *Properties* > *Security* auf die Operationsrechte (*permissions*) für jeden einzelnen Nutzer oder jede Gruppe auf diese Datei. Initial sind die Gruppen und Nutzernamen, die auf dieser Datei operieren können, relativ unspektakulär, da lediglich der Nutzer, der die Datei erstellt hat, sowie das SYSTEM als auch alle Administratoren Berechtigungen besitzen. Um dies zu ändern, editieren wir die Benutzer und Gruppen über *Edit*, wodurch sich ein zweites Fenster öffnet. In dem zweiten Fenster können wir über *Add* Benutzer zur Liste der Gruppen und Benutzer hinzufügen oder aber auch herauslöschen (*Remove*). Erfreulicherweise unterstützt die Eingabe eine Suchfunktion, die über das gesamte Benutzerverzeichnis sucht, sodass wir nur den Namen des Benutzers oder der Gruppe eingeben müssen und mittels *Enter* die Verknüpfung hinzugefügen. Unter der Liste mit Gruppen und Benutzern befindet sich ein weiteres Feld, in dem wir die Zugriffsberechtigungen eines Benutzers oder einer bestimmten Gruppe ändern können. Zum Beispiel fügen wir der Liste die Benutzer *Test1* und *Test2* sowie die Gruppe *Testgroup13* hinzu. *Test1* bekommt Lese- und Schreibberechtigungen, wohingegen *Test2* lediglich nur eine Leseberechtigung erhält. Der *Gruppe13* räumen wir eine Leseberechtigung ein. Beim durchführen eines Zugriffstests mit den jeweiligen Benutzern treffen genau diese Einstellungen zu: *Test1* darf sowohl lesen als auch schreiben, *Test2* darf nur lesen, aber nicht schreiben, *Test3* darf, aufgrund der Gruppen-Berechtigung, lesen. Die Gruppen-Berechtigung greift nicht für Benutzer *Test1*, da die Zugriffsberechtigungserteilung auf den Benutzer stärker ist als die der Gruppe.

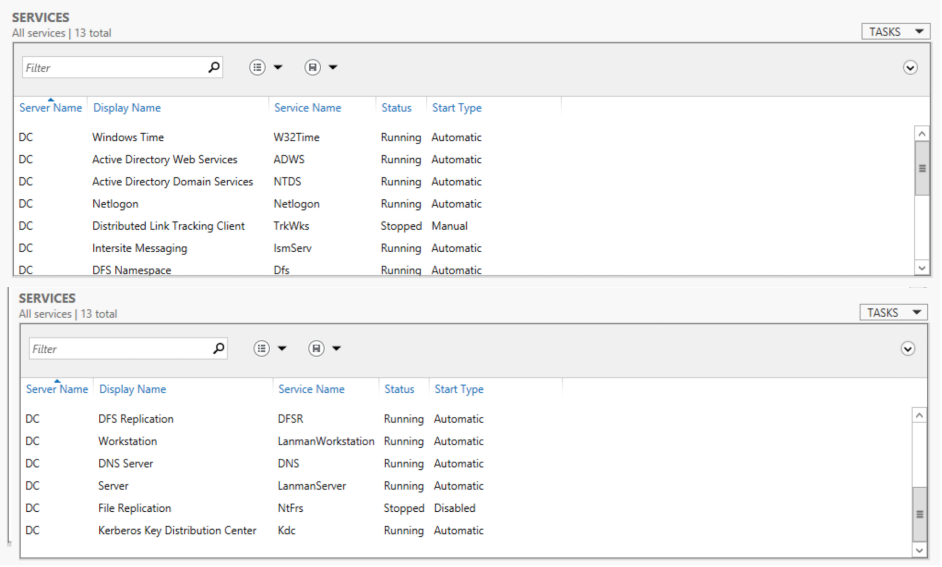

Abbildung 1: Services, von denen AD DS abhängt

Fahren wir nun weiter mit dem *Domain Controller* fort: Der *Domain Controller* ist eine Rolle auf dem Windows Server, der dazu dient Informationen über Benutzerkonten zu speichern, Benutzer zu authentifizieren und Sicherheitsregeln einer Windows-Domäne durchzusetzen.

Computer innerhalb einer Domäne, die als Server agieren, können eine von zwei Rollen innehaben: Entweder sie haben die Rolle *Member Server* oder aber *Domain Controller* [4]. Der *Domain Controller* bildet somit das Herzstück des *Active Directory Domain Service* in einem Netzwerk. Alle weiteren Services, von denen *Active Directory Domain Services* abhängt, können in den Eigenschaften des AD DS gefunden werden (siehe Abbildung 1). Darunter zählen Beispielsweise der DNS Server (wird zur Identifikation von Domain Controllern benötigt)[5], File Replication Service (dient zur Replikation von System-Policies) [6], Intersite Messaging, Key Distribution Server, Net Logon (benutzt DNS Server zur Unterstützung, um eine Registrierung von Domain Controller im DNS Domain Namespace zu ermöglichen) [7].

1.1 Installation des Domain Controllers

Bei der Konfiguration des AD Domain Services werden wird durch einen Wizard geleitet.

1. Deployment Configuration

   (a) In diesem Schritt Legen wir einen neuen Forest an und geben hierfür den Namen der Root-Domäne ein: in unserem Fall ist es die Root Domäne `acslab.local` (siehe Abbildung 2).

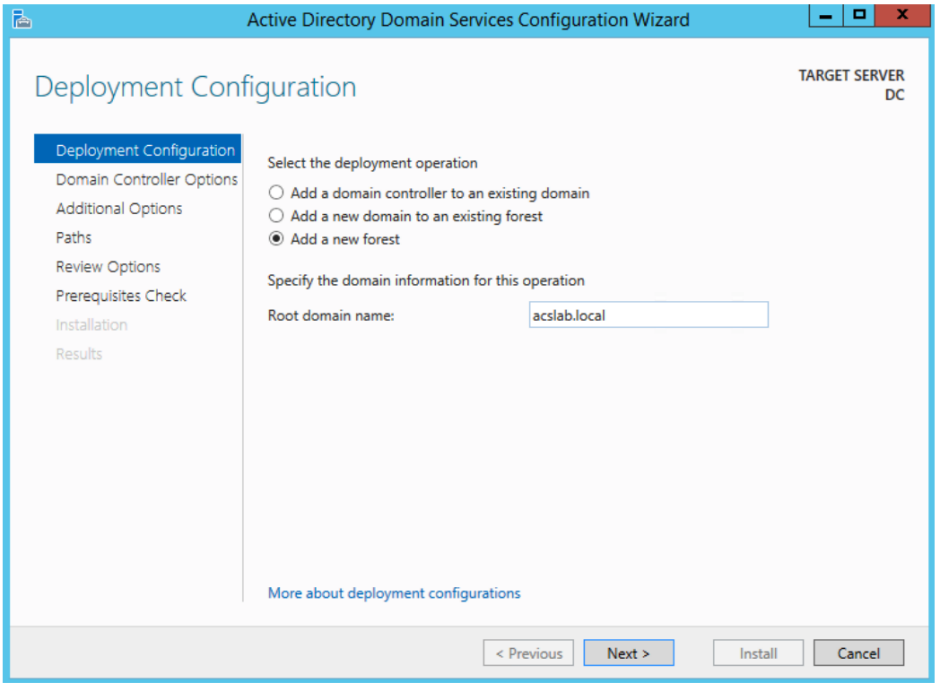

Abbildung 2: Die Deployment-Konfiguration ist der erste Schritt zur Installation des Domain Controller

2. Additional Options

[4] `http://www.webopedia.com/TERM/D/domain_controller.html`

[5] `https://docs.microsoft.com/de-de/azure/active-directory-domain-services/active-directory-ds-admin-guide-administer-dns`

[6] `https://technet.microsoft.com/en-us/library/cc978206.aspx`

[7] `http://www.windowsnetworking.com/kbase/WindowsTips/WindowsServer2008/AdminTips/ActiveDirectory/FunctionsOfNetLogonServiceOnDomainControllers.html`

(a) Da wir im Forest die Domäne `acslab.local` angelegt haben, ist in diesem Schritt die NetBIOS-Domäne ebenfalls `ACSLAB`. Die restlichen Vorkonfigurationen stimmen überein und müssen nicht weiter angepasst werden, sodass wir die Installation mit *Install* abschließen können (siehe Abbildung 3).

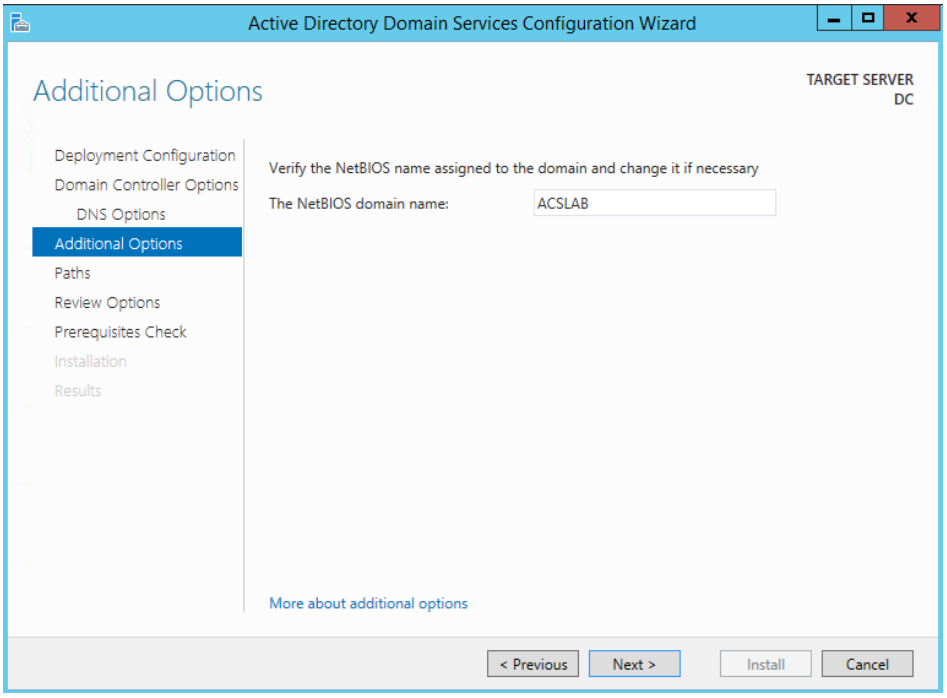

Abbildung 3: Additional Options für den Active Directory Domain Service

2 Verzeichnissynchronisation mit Azure Active Directory Connect

In diesem Abschnitt wollen wir *Azure AD Connect* installieren, um somit lokale Verzeichnisse in Azure Active Directory (Azure AD) zu integrieren. Das hat vor allem den Vorteil, dass wir unseren Benutzern eine Identität für alle in Azure AD integrierten Office-, Azure- und SaaS-Anwendungen bereitstellen können [8].

2.1 Installation von Azure AD Connect

Zur Installation von Azure AD Connect laden wir das Tool unter `https://www.microsoft.com/en-us/download/details.aspx?id=47594` herunter. In den meisten Fällen besitzt man eine autorisierte lokale Domäne und kann Azure AD Connect über das Azure Portal Dashboard herunterladen und einfach installieren. In unserem Fall handelt es sich jedoch um eine Testumgebung (und nicht um eine Produktionsumgebung), sodass wir das Tool unter der zuvor genannten offiziellen URL von Microsoft herunterladen müssen. Auch hier wird man wieder über einen Wizard durch die Installation geleitet.

1. Dadurch, dass wir allerdings noch keinen Administrator zu unserer Domäne hinzugefügt haben, lege wir zunächst einen globalen Administrator im Azure Portal Dashboard an und vergeben ihm die Rolle *Global Administrator*.

[8] `https://docs.microsoft.com/de-de/azure/active-directory/connect/active-directory-aadconnect`

2. Nachdem wir einen Administrator angelegt haben, können wir im Installationsschritt *Connect to Azure* dessen Credentials eingeben und uns mit Azure AD verbinden.

3. Unter *Domain/OU Filtering* geben wir im Verzeichnis unseren Forest-Namen *acslab.local* ein und wählen die Synchronisation mit allen Domänen und OUs aus.

4. In unseren On-Premises Verzeichnissen sollen Benutzer nur einmal entlang aller Verzeichnisse repräsentiert werden und so auch identifiziert werden.

5. Das Filtering wird so gesetzt, dass alle Benutzer und Geräte entlang unseres Forest synchronisiert werden.

6. Abschließend können wir auswählen, dass während der Installation/Konfiguration alle Services auf dem Server synchronisiert werden.

Die größten Herausforderungen die sich hierbei ergaben, war zu erkennen welche Domäne als lokale Domäne eingetragen werden soll und welcher Nutzer sich mit Azure AD verbinden muss. Dabei kann man aber relativ schnell erkennen, dass der vorherangelegte Forest als Root Domäne eingetragen werden kann und im Azure Portal ein neuer Benutzer mit globalen Administratorrechten angelegt werden muss. Ist die Installation abgeschlossen und wurden alle Services automatisch während des Installationsprozesses synchronisiert, kann über das Azure Portal eingesehen werden, dass alle Benutzer-Konten aus dem lokalen Verzeichnis angelegt wurden. Vorher tauchten diese nämlich nicht im Azure Portal auf.
Möchte man jetzt weitere Benutzer in der Domäne anlegen geht man nicht wie vorher über das lokale Server Managament sondern dieses mal über die Active Directory. Nennen wir einen neuen Benutzer Test4, geben ihm ein Standard passwort und fügen ihn der Domäne `acslab` hinzu (vergleiche dazu Abbildung 4. Im Azure Portal ändert sich allerdings zunächst nichts. Das ist auch klar, da wir noch keine Synchronisation angestoßen haben. Nachdem wir den Befehl `Start-ADSyncSyncCycle -PolicyType Delta` in der Powershell eingeben (Abbildung 5), taucht der neue Benutzer im Azure AD auf.

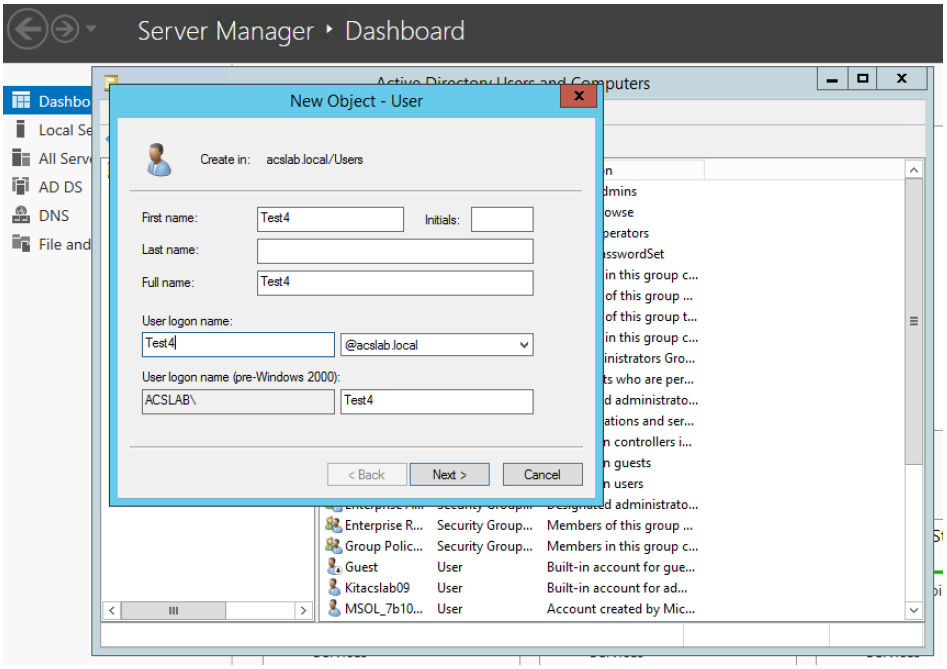

Abbildung 4: Hinzufügen eines Benutzers Test4 im AD.

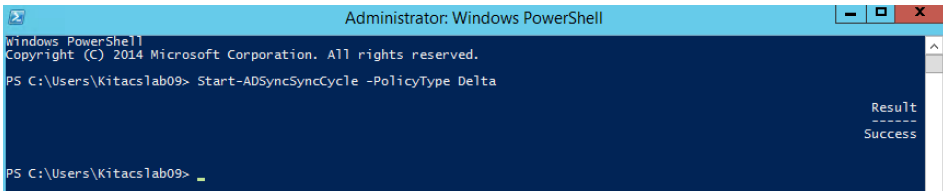

Abbildung 5: Komando zum Starten der Synchronisation über die Powershell.

3 Fazit und Diskussion

Über wenige Konfigurationsschritte und Installationen kann mittels Microsoft Azure eine gute und für den allgemeinen Anwendungsfall angemesse Lösung zur Synchronisation von Verzeichnissen ermöglicht werden.
Einer der größten Vorteile, der sich durch die Integration lokaler Verzeichnisse in Azure AD ergibt, ist, dass Benutzer für Zugriffe auf Cloud-Dienste wie Office 365 als auch auf lokale Ressourcen nur eine einzige Identität benötigen. Somit müssen sie sich auch nur ein Passwort merken, das wiederum dem Vergessen von Passwörtern entgegenwirkt. Ebenfalls erleichtert die Synchronisation den administrativen Aufwand, da eine konsistente Passwort-Policy leichter entworfen werden kann und keine User-Workstations angepasst werden müssen.
Nachteile, die sich allerdings mit der Passowort-Synchronisation und der Identitätsverwaltung ergeben sind zum einen, dass dabei auch die Wahrscheinlichkeit steigt, dass Passörter kompromittiert werden. Damit kann derjenigen mit kompromittierten Passwort auf Systeme zugreifen, die mit dem Passwort synchronisiert sind. Es gibt immer noch Systeme, die eine bi-direktionale Passwortsynchronisation nicht unterstützen. Weiterhin greifen auch Passwort-Policies längst nicht auf allen System wie es die Theorie vorsieht. Somit wird eine zentrale Verwaltung der Passwort-Policies ineffizient, da er mit einem größerer Implementierungsaufwand verbunden ist. Hinzukommt, dass die Synchronisation von Identitäten leider nicht eine sichere und prüfbare Lösung für administrative, super-user oder privilegierte Identitäten bereitstellt.[9]

Literatur

[1] https://www.microsoft.com

[2] https://de.wikipedia.org/wiki/Microsoft_Azure

[3] http://searchwindowsserver.techtarget.com/definition/Microsoft-Active-Directory-Domain-Services-AD-DS

[4] http://www.webopedia.com/TERM/D/domain_controller.html

[5] https://docs.microsoft.com/de-de/azure/active-directory-domain-services/active-directory-ds-admin-guide-administer-dns

[6] https://technet.microsoft.com/en-us/library/cc978206.aspx

[7] http://www.windowsnetworking.com/kbase/WindowsTips/WindowsServer2008/AdminTips/ActiveDirect

[8] https://docs.microsoft.com/de-de/azure/active-directory/connect/active-directory-aadconnect

[9] https://www.rsconnect.net/en/articles/pros-cons-active-directory-password-synchronisation/

[9] https://www.rsconnect.net/en/articles/pros-cons-active-directory-password-synchronisation/