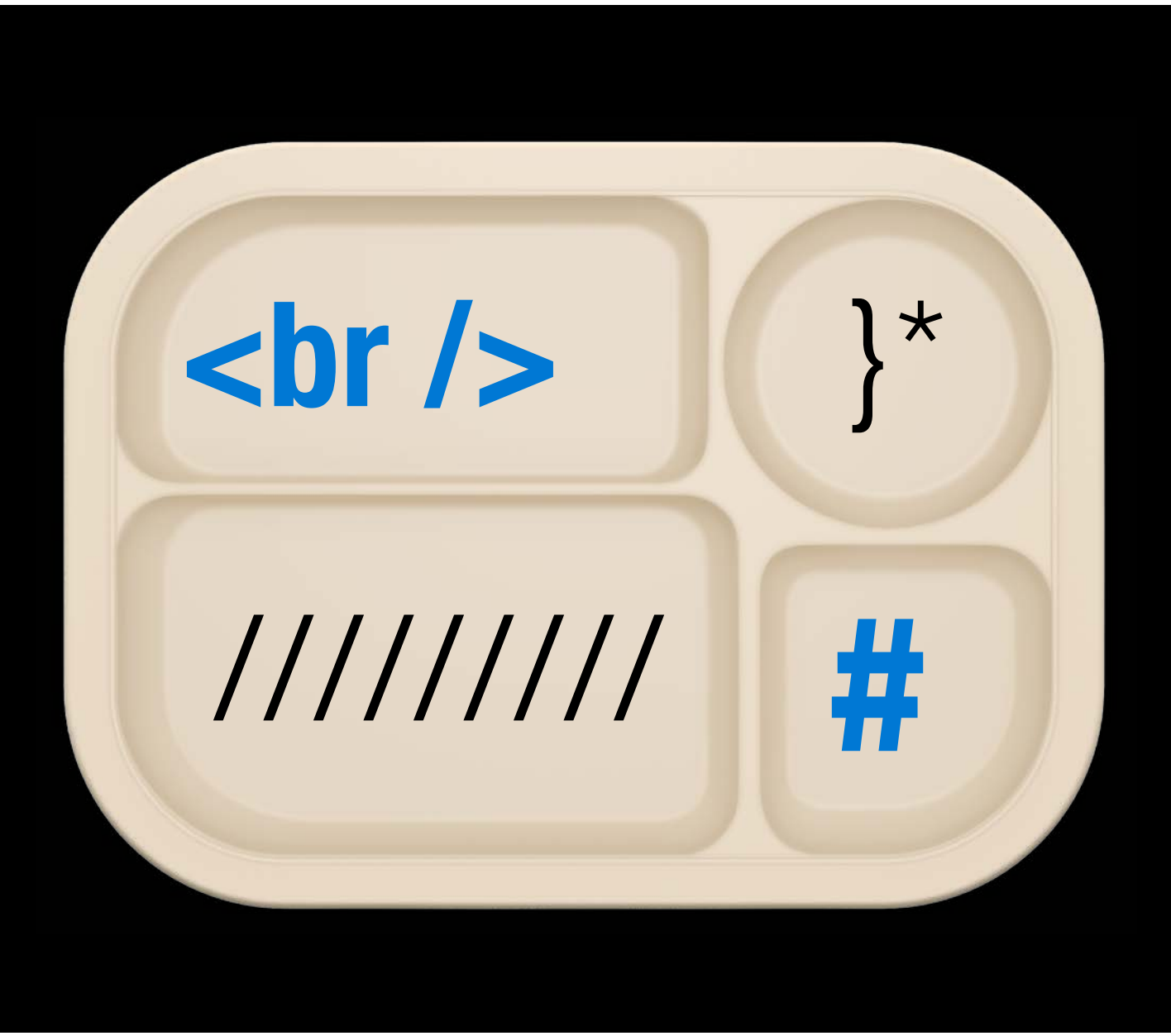
2. Auflage
Iain Foulds

NUTZEN SIE IHRE MITTAGSPAUSE UM MEHR ÜBER AZURE ZU ERFAHREN



Microsoft Azure

Holen Sie sich Hilfe für Ihr Projekt.

Vertriebsteam kontaktieren >

startHere(Azure);

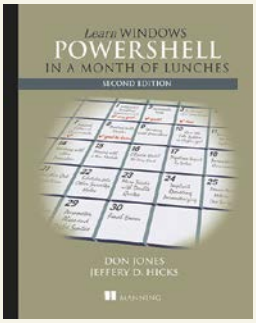
Gehen Sie 21 Azure-Lektionen durch

Nutzen Sie die Lektionen in diesem E-Book, um sich eine solide Grundlage für Azure aufzubauen. Melden Sie sich für ein kostenloses Azure-Konto an und nutzen Sie Ihre Gutschrift über 200 USD, um die Übungen abzuschließen. Fahren Sie mit Ihrem Konto fort, und nutzen Sie12 Monate lang kostenlose, beliebte Dienste und über 25 dauerhaft kostenfreie Dienste.

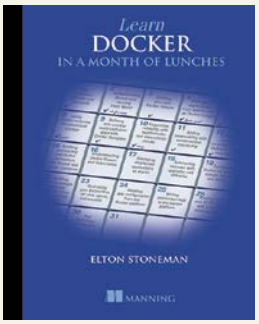
Jetzt kostenlos starten

SPAREN SIE 40 % BEI BÜCHERN UND VIDEOS VON MANNING!

Manning veröffentlicht hochwertige Bücher und Videos für Technikprofis wie Sie. Verwenden ***Sie diesen exklusiven Rabattcode, um 40 % auf alle eBooks, pBooks, MEAPs und liveVideo-Kurse auf manning.com,*** *zu erhalten, einschließlich dieser ausgewählten Titel. Geben Sie bei Ihrer Bestellung einfach* ***azuremsft2*** *in das Feld für den Aktionscode ein.*



Nutzen Sie Ihre Mittagspause, um mehr über Windows PowerShell zu erfahren
von Don Jones und Jeffery Hicks
Dezember 2016, 384 Seiten



Nutzen Sie Ihre Mittagspause, um mehr über Docker zu erfahren
Elton Stoneman
Sommer 2020, 530 Seiten

Mehr Bücher aus der Mittagspausen-Reihe

Nutzen Sie Ihre Mittagspause, um mehr über die Windows PowerShell zu erfahren, 3. Ausgabe
Nutzen Sie Ihre Mittagspause, um mehr über Docker zu erfahren
Nutzen Sie Ihre Mittagspause, um mehr über die dbatools zu erfahren
Nutzen Sie Ihre Mittagspause, um mehr über PowerShell, Linux- und macOS-Edition zu erfahren
Nutzen Sie Ihre Mittagspause, um mehr über PowerShell Scripting zu erfahren
Nutzen Sie Ihre Mittagspause, um mehr über Linux zu erfahren
Nutzen Sie Ihre Mittagspause, um mehr über die Amazon Web Services zu erfahren
Nutzen Sie Ihre Mittagspause, um mehr über die Cisco-Netzwerkadministration zu erfahren

*Bücher für Microsoft-Entwickler*innen und IT-Profis*

Azure Data Engineering
Prinzipien, Praktiken und Muster der entkoppelten Einstreuung
Mikroservices in .NET Core
.NET Core in der Praxis
Microservice Security in der Praxis
Parallelität in .NET
Reaktive Anwendungen mit Akka.NET
ASP.NET Core in der Praxis
Entity Framework Core in der Praxis
C# im Detail, 4. Ausgabe
Funktionale Programmierung in C#
Kubernetes in der Praxis
Knative in der Praxis
Bootstrapping Microservices mit Docker, Kubernetes und Terraform
Core Kubernetes
GitOps und Kubernetes
Docker in der Praxis, 2. Ausgabe
Docker in der Praxis, 2. Ausgabe
Docker in Motion
OpenShift in der Praxis
Cloud Native Patterns

Lesen Sie Bücher von Manning KOSTENFREI auf liveBook

Die liveBook-Plattform von Manning bietet ein komfortables, flexibles Online-Leseerlebnis. Sie erhalten ***KOSTENFREIEN Vollzugriff für fünf Minuten*** *am Tag für jedes Manning-Buch. Vorteile von liveBook*

- *Stellen Sie Fragen, teilen Sie Code und Beispiele, und interagieren Sie mit anderen Lesern im liveBook-Forum.*
- *Volltextsuche in allen Manning-Büchern – auch wenn Sie sie noch nicht besitzen.*
- *Registrieren Sie sich für ein kostenfreies liveBook-Konto unter livebook.manning.com.*

Sie können Ihre KOSTENFREIEN fünf Minuten auf jede erdenkliche Weise nutzen – starten und stoppen Sie den Timer, springen Sie zwischen den Büchern hin und her, und probieren Sie die interaktiven Übungen aus. Melden Sie sich einfach an, und sehen Sie sich unverbindlich um.

Lob für die 1. Ausgabe

Aus der 1. Ausgabe von *Nutzen Sie Ihre Mittagspause, um mehr über Azure zu erfahren* von Iain Foulds:

„Ein unglaublich informatives Buch, um in einem Monat sowohl grundlegende als auch fortgeschrittene Azure-Konzepte zu lernen!“

– Sushil Sharma, Galvanize

„Microsoft Azure wird schnell zu einem führenden Produkt im Bereich Public Cloud. Dank der Übungen in diesem Buch wird Ihnen der Einstieg in diese Technologie schnell gelingen.“

– Michael Bright, Developer Advocate, freiberuflicher Berater

„Ausgezeichnete Einführung zu Azure mit vielen praktischen Beispielen. Deckt ein breites Spektrum aktueller Themen ab.“

– Sven Stumpf, ING-DiBa AG

„Azure ist wie ein Ozean. Dieses Buch hält Sie über Wasser, indem es die beste Methode bietet, wie Sie in Ihrer Mittagspause anhand von praktischen Übungen und Beispielen mehr über die Plattform erfahren können.“

– Roman Levchenko, Microsoft MVP

*„Alles, was vielbeschäftigte Entwickler*innen brauchen, um den Einstieg in Azure zu finden.“*

– Rob Loranger, freiberuflicher Entwickler

„Eine großartige Möglichkeit, die Bandbreite der Azure-Angebote zu verstehen – mit einem prägnanten, auf Aktivitäten ausgerichteten Ansatz.“

– Dave Corun, Avanade

„Das umfassendste Buch zu Azure, das ich gefunden habe, um meine akademischen Projekte zu entwickeln!“

– Marco Giuseppe Salafia, PhD-Student, Università degli Studi di Catania

*„Dies ist das wichtigste Buch zur Azure-Plattform. Es ist gut organisiert, gründlich und umfassend. Beginnend mit den Grundlagen führt es die Leser*innen durch die Entwicklung immer komplexer werdender Konfigurationen mit der Azure-Plattform, um Skalierbarkeit, hohe Leistung und Redundanz für gehostete Anwendungen und Dienste zu bieten. Dieses Buch fungiert sowohl als Tutorial für Anfänger als auch als Referenz für erfahrene Benutzer*innen.“*

– Robert Walsh, Excalibur Solutions

Nutzen Sie Ihre Mittagspause, um mehr über Azure zu erfahren

2. Ausgabe

IAIN FOULDS

MANNING
SHELTER ISLAND

Online- und Bestellinformationen zu diesen und anderen Büchern von Manning finden Sie unter www.manning.com. Der Verlag bietet Rabatte für dieses Buch, wenn er in großer Stückzahl bestellt wird. Weitere Informationen erhalten Sie von

Spezielle Vertriebsabteilung
Manning Publications Co.
20 Baldwin Road
Postfach 761
Shelter Island, NY 11964
E-Mail: orders@manning.com

© 2020 by Manning Publications Co. Alle Rechte vorbehalten.

Ohne vorherige schriftliche Genehmigung des Verlags darf kein Teil dieser Publikation in irgendeiner Form oder auf irgendeine Weise, sei es elektronisch, mechanisch, durch Fotokopie oder mit sonstigen Mitteln, vervielfältigt, in einem Datenabfragesystem gespeichert oder übertragen werden.

Viele der Bezeichnungen, die von Herstellern und Verkäufern zur Unterscheidung ihrer Produkte verwendet werden, werden als Marken beansprucht. Wenn diese Bezeichnungen im Buch erscheinen und Manning Publications von einem markenrechtlichen Anspruch Kenntnis hatte, wurden die Bezeichnungen mit dem ersten Buchstaben oder allen Buchstaben als Großbuchstaben gedruckt.

♾ In Anbetracht der Bedeutung der Erhaltung des Geschriebenen ist es die Richtlinie von Manning, die von uns veröffentlichten Bücher auf säurefreiem Papier drucken zu lassen, und wir unternehmen alle Anstrengungen, um dieses Ziel zu erreichen. In Anbetracht unserer Verantwortung zur Erhaltung der Ressourcen unseres Planeten werden die Bücher von Manning auf Papier gedruckt, das zu mindestens 15 Prozent recycelt und ohne den Einsatz von elementarem Chlor verarbeitet wird.

Manning Publications Co.
20 Baldwin Road
Postfach 761
Shelter Island, NY 11964

Akquisitions-Editor:	Mike Stephens
Entwicklungs-Editorin:	Frances Lefkowitz
Editor für technische Entwicklung:	Karsten Strøbaek
Review-Editor:	Aleksandar Dragosavljevic
Produktions-Editor:	Anthony Calcara
Grafik-Editorin:	Jennifer Houle
Redakteurin:	Kathy Simpson
Korrekturleserin:	Katie Tennant
Technische Schlussredaktion	Karsten Strøbaek
Schriftsetzerin:	Marija Tudor

ISBN 9781617297625 Deckblattdesignerin: Leslie Haimes
Gedruckt in den USA.

An die ABCs meines Lebens:
Abigail, Bethany und Charlotte

Inhalt

Vorwort—xv
Danksagungen—xvi
Über dieses Buch xvii
Informationen zum Autor xxi

Teil 1: Wichtige Azure-Dienste 1

1 *Vorbereitung 3*

1.1 Ist dieses Buch für Sie geeignet? 4

1.2 Hinweise zur Verwendung dieses Buches 4

Die Hauptkapitel 4 ▪ Jetzt testen 5 ▪ Praktische Übungen 5
Quellcode und Zusatzmaterialien 5

1.3 Erstellen Ihrer praktischen Übungsumgebung 5

Erstellen eines kostenlosen Azure-Kontos 5 ▪ Praktische
Bonusübung: Erstellen eines kostenlosen GitHub-Kontos 7

1.4 Eine kleine Hilfestellung 8

1.5 Verstehen der Azure-Plattform 8

Virtualisierung in Azure 11 ▪ Verwaltungstools 12

2 *Erstellen einer virtuellen Maschine 14*

2.1 Grundlagen zur Konfiguration virtueller Maschinen 15

Regionen und Verfügbarkeitsoptionen 15 ▪ VM-Images 16
VM-Größen 17 ▪ Azure Storage 18 ▪ Virtuelle Netzwerke 20

2.2 Erstellen eines SSH-Schlüsselpaars für die Authentifizierung 20
2.3 Erstellen einer VM über Ihren Webbrowser 23
2.4 Verbindungsherstellung zur VM und Installation des Webservers 25
Verbindung zur VM mit SSH 25 ▪ Installation der Webserver 27
2.5 Zulassen, dass Webverkehr die VM erreicht 28
Erstellen einer Regel zum Zulassen von Webverkehr 28 ▪ Der Webserver in Aktion 29
2.6 Übung: Erstellen einer Windows-VM 30
2.7 Bereinigen von Ressourcen 30
2.8 Houston, wir haben ein Problem 31

3 *Azure Web Apps 33*
3.1 Azure Web Apps – Übersicht und Konzepte 34
Unterstützte Sprachen und Umgebungen 34 ▪ Staging unterschiedlicher Versionen mit Bereitstellungsslots 35 App Service-Pläne 35
3.2 Erstellen einer Web App 37
Erstellen einer grundlegenden Web App 37 ▪ Bereitstellen einer HTML-Beispielseite 39
3.3 Anzeigen von Diagnose-Log-Dateien 42
3.4 Übung: Erstellen und Nutzen eines Bereitstellungsslots 44

4 *Einführung zu Azure Storage 47*
4.1 Verwaltete Datenträger 47
Betriebssystemdatenträger 48 ▪ Temporäre Festplatten und Datenträger 50 ▪ Datenträger-Caching-Optionen 50
4.2 Hinzufügen von Datenträgern zu einer VM 51
4.3 Azure Storage 52
Table Storage 53 ▪ Queue Storage 55 ▪ Storage-Verfügbarkeit und -Redundanz 56
4.4 Übung: Erkunden von Azure Storage 57
Fokus auf die VM 57 ▪ Fokus auf Entwickelnde 57

5 **Grundlagen zu Azure Networking 58**
5.1 Virtuelle Netzwerkkomponenten 58
Virtuelle Netzwerke und Subnetze 60 ▪ Virtuelle Netzwerkschnittstellenkarten 61 ▪ Öffentliche IP-Adresse und DNS-Auflösung 62

5.2 Sicherung und Steuerung von Datenverkehr mit Netzwerksicherheitsgruppen 64

Erstellen einer Netzwerksicherheitsgruppe 65 Zuweisen einer Netzwerksicherheitsgruppe zu einem Subnetz 66 ▪ Erstellen von Filterregeln für Netzwerksicherheitsgruppen 68

5.3 Entwickeln einer Beispielwebanwendung mit sicherem Datenverkehr 68

Erstellen von Netzwerkverbindungen für den Remote-Zugriff 68Erstellen von VMs 69 ▪ Verwenden des SSH-Agents für die Verbindung zu Ihren VMs 71

5.4 Übung: Installation und Testen des LAMP-Webservers 72

Teil 2: Hochverfügbarkeit und Skalierung 73

6 Azure Resource Manager 75

6.1 Der Ansatz von Azure Resource Manager 76

Design für den Lebenszyklus der Anwendung 76 Schützen und Kontrollieren von Ressourcen 78 Schutz von Ressourcen durch Sperrung 80 Verwalten und Gruppieren von Ressourcen mit Tags 81

6.2 Azure Resource Manager-Vorlagen 82

Erstellen und Verwenden von Vorlagen 83 Erstellen von Mehrlingen eines Ressourcentyps 85 Tools für das Erstellen eigener Vorlagen 86 Speichern und Verwenden von Vorlagen 87

6.3 Übung: Bereitstellen von Azure-Ressourcen mit einer Vorlage 88

7 Hochverfügbarkeit und Redundanz 90

7.1 Die Notwendigkeit von Redundanz 90

7.2 Infrastrukturredundanz mit Verfügbarkeitszonen 92

Erstellen von Netzwerkressourcen über eine Verfügbarkeitszone hinweg 94 Erstellen von VMs in einer Verfügbarkeitszone 95

7.3 VM-Redundanz mit Verfügbarkeitsgruppen 96

Fehlerdomänen 97 Update-Domänen 97 ▪ Verteilen von VMs in einer Verfügbarkeitsgruppe 98 ▪ Anzeigen der Verteilung von VMs in einer Verfügbarkeitsgruppe 101

7.4 Übung: Bereitstellen von hochverfügbaren VMs mit einer Vorlage 102

8 Anwendungen mit Lastenausgleich 106

8.1 Azure Load Balancer-Komponenten 106

Erstellen eines Frontend-IP-Pools 108 ▪ Erstellen und Konfigurieren von Integritätstests 110 ▪ Definieren der Datenverkehrsverteilung mit Load Balancer-Regeln

112 ▪ Weiterleitung von direktem Datenverkehr mit NAT-Regeln 114 ▪ Zuweisen von Gruppen von VMs zu Backend-Pools 116

8.2 Erstellen und Konfigurieren von VMs mit dem Load Balancer 119

8.3 Übung: Anzeigen von Vorlagen vorhandener Bereitstellungen 122

9 Skalierbare Anwendungen 124

9.1 Gründe für skalierbare, zuverlässige Anwendungen 124

Vertikales Skalieren von VMs 125 ▪ Vertikales Skalieren von Web Apps 127 ▪ Horizontales Skalieren von Ressourcen 128

9.2 VM-Skalierungsgruppen 129

Erstellen einer VM-Skalierungsgruppe 131 Erstellen von Regeln für die automatische Skalierung 133

9.3 Skalieren einer Web App 136

9.4 Übung: Installation von Anwendungen in Ihrer Skalierungsgruppe oder der Web App 139

VM-Skalierungsgruppen 139 ▪ Web Apps 140

10 Globale Datenbanken mit Cosmos DB 141

10.1 Was ist Cosmos DB? 141

Strukturierte Datenbanken (SQL) 142 ▪ Unstrukturierte Datenbanken (NoSQL) 142 ▪ Skalieren von Datenbanken 143 Mit Cosmos DB alles zusammenbringen 144

10.2 Erstellen eines Cosmos DB-Kontos und einer Datenbank 145

Erstellen einer Datenbank unter Cosmos DB und Befüllung mit Daten 145 Versehen einer Cosmos DB-Datenbank mit globaler Redundanz 149

10.3 Zugriff auf global verteilte Daten 152

10.4 Übung: Einsatz einer Web App, die Cosmos DB nutzt 156

11 Verwalten von Netzwerkverkehr und Routing 158

11.1 Was ist Azure DNS? 158

11.2 Übertragen einer echten Domain auf Azure DNS 160

11.3 Globales Routing und Auflösung mit Traffic Manager 162

Erstellen von Traffic Manager-Profilen 164 ▪ Globale Verteilung des Datenverkehrs auf die nächste Instanz 167

11.4 Übung: Bereitstellung von Webanwendungen, um Traffic Manager in Aktion zu sehen 174

12 *Überwachung und Fehlerbehebung 175*

12.1 VM-Startdiagnose 175

12.2 Leistungsdaten und Warnmeldungen 178

Anzeigen der Leistungsdaten mit der VM-Diagnoseerweiterung 178 ▪ Erstellen von Benachrichtigungen für Leistungsbedingungen 181

12.3 Azure Network Watcher 182

IP-Datenflussüberprüfung 183 ▪ Anzeigen von geltenden NSG-Regeln 184 ▪ Erfassen von Netzwerkpaketen 186

12.4 Übung: Erstellen von Leistungswarnungen 188

Teil 3: Sicherheit als Standard..... 189

13 *Sicherung, Wiederherstellung und Replikation 191*

13.1 Azure Backup 191

Richtlinien und Datenaufbewahrung 193 ▪ Backup-Pläne 196 ▪ Wiederherstellen einer VM 198

13.2 Azure Site Recovery 201

13.3 Übung: Konfigurieren einer VM für Site Recovery 204

14 *Datenverschlüsselung 206*

14.1 Was ist Datenverschlüsselung? 206

14.2 Verschlüsselung ruhender Daten 208

14.3 Storage Service Encryption 209

14.4 VM-Verschlüsselung 211

Speicherung der Schlüssel in Azure Key Vault 211 ▪ Verschlüsseln einer Azure VM 213

14.5 Übung: Verschlüsseln einer VM 214

15 *Sicherung von Daten mit Azure Key Vault 216*

15.1 Die Sicherung von Daten in der Cloud 216

Softwaretresore und Hardwaresicherheitsmodule 217 ▪ Erstellen eines Schlüsseltresors und eines Geheimnisses 219

15.2 Verwaltete Identitäten für Azure-Ressourcen 221

15.3 Erhalten eines Geheimnisses aus einer VM mit einer Identität aus einem verwalteten Service 224

15.4 Erstellen und Einspeisen von Zertifikaten 229
15.5 Übung: Konfigurieren eines sicheren Webservers 232

16 *Azure Security Center und Updates 234*
16.1 Azure Security Center 234
16.2 Zeitlich beschränkter Zugriff 237
16.3 Azure Update Management 241
Kombinierte Azure-Verwaltungsdienste 243 Überprüfen und Anwenden von Updates 245
16.4 Übung: Aktivieren von JIT und Updates für eine Windows-VM 249

TEIL 4: DAS COOLE ZEUG 251

17 *Machine Learning und künstliche Intelligenz 253*
17.1 Überblick zu und Beziehung von KI und ML 254
*Künstliche Intelligenz 254 ▪ Machine Learning 255 ▪ Zusammenbringen von KI und ML 256 ▪ Azure ML-Tools für Datenwissenschaftler*innen 257*
17.2 Azure Cognitive Services 259
17.3 Erstellung eines intelligenten Bots zur Unterstützung bei Pizzabestellungen 260
Erstellen eines Azure Web App Bots 260 ▪ Sprache und Verständnis von Absichten mit LUIS 261 ▪ Erstellen und Betreiben eines Web App Bots mit LUIS 264
17.4 Übung: Hinzufügen von Kanälen für die Botkommunikation 267

18 *Azure Automation 269*
18.1 Was ist Azure Automation? 269
Erstellen eines Azure Automation-Kontos 271 ▪ Azure Automation-Assets und -Runbooks 272
18.2 Azure Automation-Beispiel-Runbook 274
Ausführen und Anzeigen von Ausgaben aus einem Beispiel-Runbook 276
18.3 PowerShell Desired State Configuration (DSC) 278
Definieren und Nutzen von PowerShell DSC und einem Azure Automation Pull Server 280
18.4 Übung: DSC mit Linux 282

19 *Azure Container 284*
19.1 Was sind Container? 284

19.2 Das Microservice-Konzept für Anwendungen 287
19.3 Azure Container-Instanzen 289
19.4 Azure Kubernetes Service 293
Erstellen eines Clusters mit Azure Kubernetes Services 294
Betrieb einer einfachen Website auf Kubernetes 295
19.5 Übung: Skalieren Ihrer Kubernetes-Bereitstellungen 298

20 *Azure und das Internet der Dinge 300*
20.1 Was ist das „Internet der Dinge“? 300
20.2 Zentrale Geräteverwaltung mit Azure IoT Hub 303
20.3 Erstellen eines simulierten Raspberry Pi-Geräts 306
20.4 Streamen von Azure IoT-Hub-Daten zu Azure Web Apps 309
20.5 Azure IoT-Komponentenprüfung 315
20.6 Übung: Ausprobieren von IoT-Beispielanwendungen 316

21 *Serverless Computing 317*
21.1 Was ist Serverless Computing? 317
21.2 Azure-Messaging-Plattformen 319
Azure Event Grid 320 ▪ Azure Event Hubs und Service Bus 321 ▪ Erstellen eines Service Bus und Integration in einen IoT Hub 322
21.3 Erstellen einer Azure Logic App 325
21.4 Erstellen einer Azure Function App zur Analyse von IoT-Gerätedaten 328
21.5 Machen Sie mit dem Lernen weiter 332
Ergänzende Lernmaterialien 333 ▪ GitHub-Ressourcen 333 ▪ Ein abschließender Gedanke 333

index 335

Vorwort

Diese zweite Ausgabe von *Nutzen Sie Ihre Mittagspause, um mehr über Azure zu erfahren,* erinnert mich daran, dass sich die Dinge sehr schnell ändern können und dass es wichtig ist, sich ständig weiterzubilden. Vorbei sind die Zeiten, in denen Sie an einem einwöchigen Kurs in Windows Server teilnehmen und das System dann problemlos jahrelang ausführen konnten, ohne viel ändern zu müssen. Das bedeutet allerdings nicht, dass die IT-Welt ein beängstigenderer Ort geworden ist, Sie müssen Cloud Computing allerdings unvoreingenommen betrachten und bereit sein, sich ständig anzupassen.

Als ich anfing, mit Azure zu arbeiten, war die Anzahl der verfügbaren Dienste überwältigend. Ich wusste, dass ich auf Sicherheit, Leistung, Redundanz und Skalierbarkeit achten sollte. Ich wusste aber nicht, wie ich über ein Jahrzehnt groß angelegter Server-Administration bei Cloud Computing anwenden sollte. Im Laufe der Zeit begann ich, mehr über die verschiedenen Azure-Dienste zu erfahren, die diese Schlüsselkomponenten bereitstellen. Diese Dienste arbeiten selten isoliert, aber ich wusste nicht, wie ich sie am besten integrieren könnte oder welchen Dienst ich für welche Aufgabe verwenden sollte. Dieses Buch bietet eine Möglichkeit, meinem früheren Selbst und vielen anderen, die einen ähnlichen Weg gegangen sind, zu erklären, wie sie die Kernleistungen von Azure schnell verstehen und zusammenwirken lassen können.

Dieses Buch hat über 350 Seiten, aber es kratzt kaum an der Oberfläche der Vielzahl von Möglichkeiten auf Azure! Um Ihnen ein solides Verständnis der für die erfolgreiche Entwicklung von Lösungen in Azure erforderlichen Konzepte zu vermitteln, musste ich mich beim Verfassen dieses Buchs für bestimmt Themen entscheiden. Das Buch befasst sich nicht mit allen 100 oder mehr Azure-Diensten, und es enthält keine ausführlichen Details über die bereitgestellten Dienste. Stattdessen konzentriert es sich auf die wichtigsten Bereiche in einigen der primären Dienste, zeigt Beispiele für eine sichere Verbindung aller Dienste und stellt die Möglichkeiten für die Entwicklung in Azure vor.

Cloud Computing unterliegt ständigen Veränderungen. Es gibt keine drei oder vierjährigen Release-Zyklen und keine großen Update-Bereitstellungen. Ich glaube, dass heute eine großartige Zeit ist, um Lösungen zu entwickeln und Code zu schreiben. Ihnen bietet sich dadurch immer eine Möglichkeit, etwas Neues zu lernen und sich weiterzuentwickeln. Ich hoffe, Sie lernen, großartige Anwendungen in Azure auszuführen und haben Spaß bei der Entdeckung aller verfügbaren Dienste.

Danksagungen

Viele Menschen hinter den Kulissen von Manning Publications haben an der Veröffentlichung dieses Buches mitgewirkt. Besonderer Dank geht an Mike Stephens, der die Vision hatte, dieses Projekt ins Leben zu rufen. Ich bedanke mich auch bei meinem Herausgeber Marjan Bace und allen Mitarbeiter*innen der Redaktion und Produktion. Mein Dank gilt den technischen Peer Reviewern unter der Leitung von Aleksandar Dragosavljevi—: Ariel Gamino, Charles Lam, Ernesto Cardenas Cangahuala, George Onofrei, Glen Thompson, Jose Apablaza, Juraj Borza, Michael Langdon, Michael Wall, Peter Kreyenhop, Rick Oller, Rob Ruetsch, Robert Walsh, and Vishal Singh. Schließlich möchte ich mich bei Karsten Strøbaek bedanken, der sowohl als technischer Editor als auch als Korrekturleser dieses Buches tätig war.

Für diese 2. Auflage möchte ich Phil Evans und Davanand Bahall für ihre Unterstützung und die Freiheit, dieses Buch zu aktualisieren, herzlich danken. Es handelte sich um ein Projekt, das ich nach Feierabend und außerhalb meiner täglichen Arbeit bei Microsoft durchführte. Trotzdem erreichte und begeisterte es viele Menschen. Vielen Dank auch an David Tolkov und Tim Teebken, durch die ich mich zu jemandem entwickeln konnte, der ein solches Buch schreiben kann. Und schau, Jean-Paul Connock, wir haben seit dem letzten Mal einen Stanley Cup gewonnen! Los Blues!

Vielen Dank an Rick Claus für seine Unterstützung bei der Erstellung einer umfassenden technischen Dokumentation in Azure und an Marsh Macy und Neil Peterson für ihre persönliche Unterstützung und Anleitung beim Schreiben der Originalversion dieses Buches. Wir müssen immer noch in diesen Schulbus einsteigen.

Über dieses Buch

Dieses Buch wurde verfasst, um Ihnen eine solide Grundlage zu vermitteln, als Entwickler*in oder IT-Ingenieur*in mit Azure erfolgreich zu sein. Sie erfahren darin sowohl etwas über Infrastructure-as-a-Service (IaaS)- als auch über Platform as a Service (PaaS)-Lösungen sowie darüber, wann Sie den jeweiligen Ansatz anwenden sollten. Beim Durcharbeiten der Kapitel dieses Buches erfahren Sie, wie Sie die Verfügbarkeit und Skalierbarkeit angemessen planen, die Sicherheit im Auge behalten und Kosten und Leistung berücksichtigen. Am Ende des Buches sollten Sie in der Lage sein, kommende Technologien wie Container und Kubernetes, künstliche Intelligenz und Machine Learning (KI + ML) und das Internet der Dinge (Internet of Things, IoT) zu integrieren.

Wenn es um die Erstellung und Ausführung Ihrer Anwendungen und Dienste geht, können Sie mit Azure das Betriebssystem, die Anwendungstools und die Plattform auswählen, mit denen Sie am besten vertraut sind. Dieses Buch behandelt hauptsächlich Nicht-Microsoft-Technologien wie Linux, Python und Node.js. Befehlsbeispiele verwenden die Azure CLI und nicht Azure PowerShell. Dies waren bewusste Entscheidungen. Damit wollte ich Ihnen demonstrieren, dass Sie bei der Verwendung von Azure nicht unbedingt Windows Server, IIS oder ASP.NET verwenden müssen.

Die Arbeit in der Cloud muss oft plattformübergreifend erfolgen, und Sie müssen kontinuierlich neue Themen erlernen. Dies ist ein weiterer Grund dafür, auch auf Nicht-Microsoft-Technologien und -Plattformen einzugehen. Ich wollte Ihnen einige dieser neuen Bereiche vorstellen, bevor Sie ihnen in der realen Praxis begegnen. Im gesamten Buch werde ich Ihnen Konzepte und Schritte näherbringen, die zur Integration von Azure-Diensten erforderlich sind. Damit möchte ich erreichen, dass Sie zwischen Plattformen oder Sprachen beliebig wechseln und dabei dieselben erworbenen Kenntnisse anwenden.

Planung

Das Buch ist in 4 Teile und 21 Kapitel gegliedert:

- Teil 1 umfasst einige der wichtigsten Elemente der Azure-Infrastruktur und Plattformdienste: Virtual Machines, Web Apps, Storage und Networking.
- Teil 2 geht der Frage nach, wie Hochverfügbarkeit und Redundanz bereitgestellt werden können: Vorlagen, Verfügbarkeitssets und -bereiche, Load Balancer, Autoskalierung, verteilte Datenbanken und Datenverkehrsrouting. Am Ende von Kapitel 12 sollten Sie über fundierte Kenntnisse in der Erstellung von leistungsstarken, verteilten Anwendungen in Azure verfügen.
- Teil 3 behandelt Sicherheitsaspekte wie Sicherung und Wiederherstellung, Verschlüsselung, Verwaltung digitaler Schlüssel und Updates. Nach Abschluss von Kapitel 16 sind Sie auf dem besten Weg, sichere und stabile Anwendungen in Azure zu erstellen.
- Zum Abschluss des Buches werden in Teil 4 auf unterhaltsame Weise neue Bereiche des Computing wie Serverless Computing und containerbasierte Anwendungen erkundet. In diesen Kapiteln werden Bereiche von Azure vorgestellt, die Ihnen einen Einblick in die Zukunft von Produktionsanwendungen bieten.

Abgesehen von Teil 4 (der die passende Überschrift „Das coole Zeug“ hat), sollten Sie versuchen, die Kapitel des Buches in der richtigen Reihenfolge durchzuarbeiten. Sie arbeiten nicht am gleichen Projekt in mehreren Kapiteln, aber jedes Kapitel baut auf den Theorien und praktischen Übungsbeispielen aus den vorherigen Kapiteln auf.

Kapitel 1 führt Sie durch die Erstellung eines kostenlosen Testkontos in Azure, mit dem Sie praktische Übungen in jedem Kapitel absolvieren können. Ich stelle Ihnen auch ein wenig mehr Hintergrundinformationen über Azure zur Verfügung und zeige Ihnen, wie Sie Hilfestellungen auf Ihrem Weg finden. Ich erwähne diese Webseite ein paar Mal in dem Buch (vielleicht bin ich ein wenig voreingenommen!), aber http://docs.microsoft.com/azure ist die beste Quelle für zusätzliche Dokumente und Unterstützung zu allen Bereichen von Azure, die für Sie von Interesse sein könnten.

Informationen zu den Beispielen und dem Quellcode

Dieses Buch enthält viele Beispiele für Quellcode, sowohl in nummerierten Verzeichnissen als auch in normalem Text. In beiden Fällen wird der Quellcode in `fetter Schriftart` formatiert, um ihn vom normalen Text zu trennen.

In vielen Fällen wurde der ursprüngliche Quellcode neu formatiert. Wir haben Zeilenumbrüche und überarbeitete Einrückungen eingefügt, um den verfügbaren Seitenplatz im Buch anzupassen. In einigen wenigen Fällen reichte selbst dies nicht aus, und die Listen enthalten Markierungen für Zeilenumbrüche (➥). Darüber hinaus werden Kommentare im Quellcode aus den Verzeichnissen entfernt, wenn der Code im Text beschrieben wird. Code-Anmerkungen sind in vielen Verzeichnissen enthalten, um wichtige Konzepte zu verdeutlichen.

Den Quellcode dieses Buches sowie begleitende Skripte, Vorlagen und unterstützende Ressourcen finden Sie unter https://www.manning.com/books/learn-azure-in-a-month-of-lunches-second-edition und im GitHub-Repository dieses Buches (https://github.com/fouldsy/azure-mol-samples-2nd-ed).

Alle praktischen Übungen können im Azure Portal und mit der Azure Cloud Shell, einer browserbasierten interaktiven Shell für den Azure CLI und Azure PowerShell, abgeschlossen werden. Es müssen keine Tools auf Ihrem Computer installiert werden, und Sie können jeden beliebigen Computer und jedes beliebige Betriebssystem verwenden, vorausgesetzt, es unterstützt einen modernen Webbrowser.

Das Azure Portal implementiert oft kleinere Änderungen. Ein Teil der Herausforderung bei der Nutzung von Cloud-Diensten besteht darin, dass die Dinge sich von einem auf den anderen Tag ein wenig ändern können.

In dieser 2. Ausgabe des Buches wird versucht, die Anzahl der Portal-Screenshots zu minimieren. Sie müssen sich aber keine Sorgen machen, wenn sich die angezeigten Bilder immer noch ein wenig von den Bildern im Buch unterscheiden. Die benötigten Parameter sind in der Regel dieselben – das Layout ist möglicherweise einfach etwas anders. Bei neuen Optionen im Portal, die ich nicht speziell in einer Übung oder einem praktischen Übung aufrufe, können Sie normalerweise die bereitgestellten Standardwerte guten Gewissens übernehmen.

Beachten Sie die Befehlsbeispiele, wenn Sie außerhalb der Azure Cloud Shell arbeiten. Windows-basierte Shells wie PowerShell und CMD behandeln Zeilenumbrüche und -fortsetzungen anders als *nix-basierte Shells wie die Azure Cloud Shell. Viele der Befehlsbeispiele gehen über mehrere Zeilen. Die Befehle werden mit einem Backslash (\) angezeigt, um anzugeben, dass der Befehl in der nächsten Zeile fortgesetzt wird, wie im folgenden Beispiel gezeigt:

```
az resource group create \
--name azuremol \
--location eastus
```

Sie müssen diese Backslash-Zeichen nicht eingeben, lange Befehle werden aber so auf Ihrem Bildschirm besser lesbar. Wenn Sie sich dafür entscheiden, mit einer Windows-Shell lokal auf Ihrem Computer zu arbeiten, können Sie einen Backtick (`) anstelle eines Backslash verwenden. Ändern Sie beispielsweise in einer PowerShell- oder CMD-Shell mit installiertem Python für Windows den vorherigen Befehl wie folgt:

```
az resource group create `
--name azuremol `
--location eastus
```

Diese Konvention mag zunächst verwirrend erscheinen, aber ich folge dieser Konvention im Buch, da die offizielle Dokumentation unter https://docs.microsoft.com/azure dieses Format verwendet. Azure CLI-Befehle, die in diesem Buch hauptsächlich verwendet werden, gehen von einer *nix-basierten Shell aus und verwenden daher ein Backslash-Zeichen. Azure PowerShell-Befehle gehen von einer Windows-basierten Shell aus und verwenden daher einen Backtick. Dieser Unterschied

macht schnell Sinn, und Sie werden feststellen, dass es einfach ist, zwischen Shells zu wechseln. Wenn Sie noch ein Neuling im Bereich der plattformübergreifenden Arbeit sind, dann kann dieser Unterschied zu einer ziemlich lustigen Angelegenheit werden!

Ich empfehle Ihnen das Windows-Subsystem für Linux (WSL), wenn Sie Windows 10 verwenden und sich mit der Azure CLI und *nix-basierten Systemen im Allgemeinen beschäftigen möchten. Details dazu finden Sie unter https://docs.microsoft.com/windows/wsl. WSL und die neuesten Verbesserungen in WSL2 bieten Ihnen ein natives Linux-Kernel-Erlebnis, während Sie Windows ausführen. Versuchen Sie nicht, sich einen vollständigen Reim auf diese Idee zu machen. Sie müssen nur wissen, dass Sie native Linux-Befehle und -Anwendungen ausführen können, ohne sich um unterschiedliche Zeilenumbrüche oder Variablendefinitionen kümmern zu müssen. Was Sie wirklich begeistern wird: PowerShell ist für .NET Core verfügbar, das auch unter Linux läuft. Sie können PowerShell unter Linux in Windows ausführen.

liveBook-Diskussionsforum

Der Kauf von *Nutzen Sie Ihre Mittagspause, um mehr über Azure zu erfahren,* beinhaltet einen kostenlosen Zugang zu einem privaten Webforum von Manning Publications. Dort können Sie Kommentare zum Buch abgeben, technische Fragen stellen und Hilfe vom Autor und anderen Benutzern erhalten. Sie können auf das Forum unter https://livebook.manning.com/book/learn-azure-in-a-month-of-lunches-second-edition/discussion zugreifen. Weitere Informationen zu den Foren und Verhaltensregeln von Manning finden Sie unter https://livebook.manning.com/discussion.

Manning ist bestrebt, für unsere Leser*innen einen Ort zu schaffen, an dem ein sinnvoller Dialog zwischen einzelnen Leser*innen sowie zwischen Leser*innen und dem Autor stattfinden kann. Es handelt sich dabei um keine Verpflichtung zu einer bestimmten Beteiligungsaktivität des Autors, dessen Beiträge zum Forum freiwillig (und unbezahlt) bleiben. Wir empfehlen Ihnen, dem Autor einige herausfordernde Fragen zu stellen, damit sein Interesse weiterhin geweckt bleibt! Das Forum und das Archiv mit früheren Diskussionen stehen auf der Website des Verlages zur Verfügung, solange das Buch im Handel erhältlich ist.

Informationen zum Autor

IAIN FOULDS ist Senior Content Developer bei Microsoft und schreibt derzeit technische Dokumentationen für Azure Active Directory. Zuvor hatte er bei Microsoft die Position eines führenden Außendiensttechnikers für Virtualisierungstechnologien wie Azure, Hyper-V und System Center Virtual Machine Manager inne. Mit über 15 Jahren Erfahrung in der IT, hauptsächlich im Bereich Prozesse und Dienste, hat er die Virtualisierung frühzeitig bei VMware eingeführt und unterstützt seit Jahren die Entwicklung und die Schulung anderer beim Thema Cloud Computing.

Iain, der ursprünglich aus England stammt, lebt seit mehr als einem Jahrzehnt in den Vereinigten Staaten. Er wohnt derzeit mit seiner Frau und zwei kleinen Kindern, denen dieses Buch gewidmet ist, etwas außerhalb von Seattle. Er ist ein Fußballfan (der Sportart, die in den USA aber unglücklicherweise als „Soccer“ bezeichnet wird) und interessiert sich auch für Eishockey und fast jede Form von Autorennen. Neben dem Computing liegen seine Interessen in den Bereichen Performance- und Oldtimerfahrzeuge sowie Flugzeugfotografie. Er ist auch fest davon überzeugt, Gitarre spielen zu können. Er ist auch ein großer Modelleisenbahn-Nerd, der regelmäßig an Shows und Veranstaltungen im gesamten Nordwesten der Pazifikküste teilnimmt und sich ehrenamtlich engagiert.

Teil 1:

Wichtige Azure-Dienste

Zum Erstellen der nächsten großen Anwendung benötigen Sie ein solides Verständnis der grundlegenden Ressourcen von Azure. Aspekte wie Speicher und Vernetzung sind vielleicht nicht die angesagtesten, doch sie sind von zentraler Bedeutung für viele Dinge, die mit Azure ausgeführt werden. Bevor Sie mit dem Einstieg in redundante, virtuelle Maschinen mit mehreren Instanzen oder Azure Web Apps beginnen können, sollten Sie die verfügbaren Optionen und Verwaltungsaufgaben für eine einzelne Instanz prüfen. Mit diesem Ansatz lernen Sie die Gemeinsamkeiten von und die Unterschiede zwischen dem IaaS-Ansatz für VMs und dem PaaS-Ansatz für Web Apps kennen. In den Kapiteln 1 bis 5 erkunden wir VMs und Web Apps sowie wichtige Storage- und virtuelle Netzwerkfunktionen.

1 Vorbereitung

Azure ist einer der größten Public Cloud Computing-Anbieter für Dienste wie virtuelle Maschinen (VMs), Container, Serverless Computing und Machine Learning. Wir werden in diesem Buch nicht alle 100 oder mehr Azure-Dienste im Detail behandeln, aber Sie erfahren mehr über die wichtigsten Dienste und Funktionen, die den Großteil der benötigten Aspekte ausmachen, um mit dem Erstellen und Ausführen von Lösungen in Azure zu beginnen. Wir werden uns ein gängiges Beispiel dafür ansehen, wie eine Web App entwickelt und ausgeführt wird, und Sie erfahren, wie Sie einige der wichtigsten Infrastruktur- und Plattformdienste nutzen können, mit denen Sie sich Ihre Arbeit erleichtern können.

Mit Azure können Sie ganz ohne Hexerei prognostizieren, wie viele Server oder wie viel Speicher Sie in den nächsten drei Jahren benötigen. Keine weiteren Verzögerungen beim Erhalt von Budgetzulassungen, kein Warten auf den Versand neuer Hardware und kein Sortieren, Installieren und Konfigurieren mehr. Machen Sie sich beim Schreiben Ihres Codes keine Gedanken mehr darüber, welche Softwareversionen oder Bibliotheken installiert sind.

Stattdessen betätigen Sie nur eine Schaltfläche und erstellen so die benötigten Ressourcen. Sie zahlen nur für jede Minute, in der diese Ressourcen ausgeführt werden, bzw. für die verwendete Menge an Speicherplatz oder Netzwerkbandbreite. Wenn Sie die Ressourcen nicht mehr benötigen, können Sie sie herunterfahren oder löschen. Wenn Sie plötzlich die Menge der Rechenleistung um den Faktor 10 erhöhen müssen, betätigen Sie eine Schaltfläche, warten einige Minuten – und fertig. Das alles wird von einer anderen Person verwaltet, sodass Sie sich auf Ihre Anwendungen und Kund*innen konzentrieren können.

1.1 Ist dieses Buch für Sie geeignet?

Die IT-Branche befindet sich bezüglich Berufsbezeichnung in einer gewissen Übergangsphase. Sie können sich selbst als IT-Profi, Softwareentwickler*in, Systemadministrator*in oder DevOps-Ingenieur*in bezeichnen. Egal, wie Sie sich nennen: Wenn Sie die wichtigsten Kompetenzen zum Entwickeln und Ausführen sicherer, hochverfügbarer Anwendungen in der Cloud erlernen möchten, sind Sie hier richtig. Allgemein ausgedrückt gehören Sie wahrscheinlich dem Bereich IT-Betrieb oder Entwicklung an. Tatsächlich aber gibt es eine Menge Überschneidungen, vor allem beim Cloud Computing. Ob Sie nun in der Entwicklung oder im Betrieb tätig sind: Es ist wichtig, die wichtigsten Infrastruktur- und Plattformdienste zu verstehen, um Anwendungen so zu entwickeln und auszuführen, dass sie optimal für Ihre Kund*innen geeignet sind.

In der 2. Ausgabe dieses Buches werden einige dieser Kernkonzepte in Azure vorgestellt, und Ihnen werden die Fähigkeiten vermittelt, die Sie zum Treffen fundierter Entscheidungen benötigen. Sie sollten über einige Vorkenntnisse mit VMs verfügen und die Grundlagen von Netzwerk- und Speicherkonzepten kennen. Sie sollten auch dazu in der Lage sein, eine grundlegende Website zu erstellen und zu verstehen,was ein SSL-Zertifikat und eine Datenbank sind. Nachdem wir die Kernprozesse behandelt haben, werfen wir einen kurzen Blick auf neue und zukünftige Technologien. Sie möchten immer einen Schritt voraus sein, wenn es um Ihre Arbeit geht. Deshalb lernen Sie etwas über Container, das Internet der Dinge, Machine Learning, künstliche Intelligenz und Serverless Computing. Sowohl selbsternannte Entwickler*innen als auch IT-Profis können hier wahrscheinlich einige neue Bereiche finden, über die sie noch etwas lernen können.

1.2 Hinweise zur Verwendung dieses Buches

Ich mag Sandwiches, deshalb ist das Mittagessen eine gute Zeit für mich, um mit cooler neuer Technik zu spielen. Egal, ob Sie ein Nachtschwärmer sind, der abends etwas Zeit übrig hat, oder ein Frühaufsteher (was ist los mit Ihnen?!), der beim Frühstück ein Kapitel durchgehen kann. Es gibt keine richtige oder falsche Zeit zu lernen. Wenn Sie etwa 45 Minuten haben, können Sie ein Kapitel lesen und die Übungen dazu abschließen. In jedem Kapitel wird etwas Neues behandelt, also nehmen Sie sich Zeit, die Lektion des jeweiligen Tages durchzugehen.

1.2.1 Die Hauptkapitel

Das Buch ist in vier Teile aufgeteilt, was praktisch ist, da jeder Monat mehr oder weniger vier Wochen hat:

- Teil 1 (Kapitel 1–5) deckt einen Teil der wichtigsten Azure-Ressourcen ab. Versuchen Sie auf jeden Fall, diese Kapitel zu lesen, um ein solides Verständnis zu erlangen. Anschließend können Sie sich auf die anderen Kapitel konzentrieren, die Sie am meisten interessieren.
- Teil 2 (Kapitel 6–12) umfasst Verfügbarkeit und Skalierbarkeit. Sie erfahren, wie Sie Ressourcen automatisch hoch- und runterskalieren, den Lastenausgleich für Datenverkehr verwalten und Wartungsereignisse ohne Ausfallzeiten abwickeln können. Wenn Sie mehr über das Ausführen hochverfügbarer Anwendungen auf globaler Ebene erfahren möchten, ist dieser Abschnitt für Sie wichtig.
- Teil 3 (Kapitel 13–16) richtet sich an die Sicherheits-Geeks. Er umfasst Dinge wie die Verschlüsselung von VMs, die Speicherung von SSL-Zertifikaten in einem sicheren Tresor und die Sicherung und Wiederherstellung Ihrer Daten.

- Teil 4 (Kapitel 17–21) umfasst eine Mischung aus interessanten Bereichen, und Sie erhalten einen Vorgeschmack darauf, was Azure für Sie und Ihre Kund*innen leisten kann. Wir beschäftigen uns mit Automatisierung, Containern, dem Internet der Dinge und Serverless Computing. Wählen Sie ein für Sie interessantes Thema, und haben Sie viel Spaß beim Lesen!

1.2.2 Jetzt testen

Möchten Sie nur lesen oder die Ärmel hochkrempeln und mit Azure experimentieren? Im ganzen Buch gibt es immer wieder kleine Aufgaben, mit denen Sie schnell etwas Neues ausprobieren können. Wenn Sie Zeit haben, versuchen Sie es. Die meiste praktische Arbeit ist am Ende des Kapitels in einer praktischen Übung zu finden. Es ist allerdings sehr sinnvoll, das Lesen zuvor zu unterbrechen, um bis zum Ende der Lektüre neue Konzepte auszuprobieren. Einige dieser Übungen leiten Sie Schritt für Schritt an. Bei anderen müssen Sie hingegen etwas mehr nachdenken und lernen, wie Sie selbst Lösungen entwickeln können – genau wie Sie es in der echten Welt tun würden.

1.2.3 Praktische Übungen

Jedes Kapitel wird mit einer praktischen Übung abgeschlossen. Bei einigen Kapiteln, wie diesem, gibt es eine praktische Übung in der Mitte des Kapitels. Anhand dieser praktischen Übungen erfahren Sie, wie alle Teile von Azure zusammenkommen, und Sie können damit beginnen, Ihr mentales Muskelgedächtnis aufzubauen. Schnappen Sie sich Ihre Tastatur und Maus, und entwickeln Sie etwas Geniales.

1.2.4 Quellcode und Zusatzmaterialien

Den Quellcode dieses Buches sowie begleitende Skripte, Vorlagen und unterstützende Ressourcen finden Sie unter https://www.manning.com/books/learn-azure-in-a-month-of-lunches-second-edition und im GitHub-Repository dieses Buches unter https://github.com/fouldsy/azure-mol-samples-2nd-ed. Darüber hinaus können Sie sich am Forum des Buches beteiligen unter https://livebook.manning.com/book/learn-azure-in-a-month-of-lunches-second-edition/discussion.

1.3 Erstellen Ihrer praktischen Übungsumgebung

Dieses Buch legt den Schwerpunkt nicht auf Konzepte und Architektur – es geht mehr um die Praxiserfahrung mit der Azure-Plattform. Um diese Praxis zu bekommen, benötigen Sie ein Azure-Konto.

1.3.1 Erstellen eines kostenlosen Azure-Kontos

Azure bietet ein kostenloses Testkonto für 30 Tage an, und Sie erhalten eine Gutschrift in Höhe von 200 US-Dollar. Diese Gutschrift sollte ausreichen, um alle Kapitel und Übungen durchzuarbeiten und dabei auch noch ein wenig zu lernen und Spaß zu haben. Es gibt viele Azure-Dienste und -Funktionen, die auch nach dem Ende Ihres Testzeitraums kostenlos bleiben.

Jetzt testen

Befolgen Sie die Schritte in diesem Abschnitt, um Ihr kostenloses Azure-Konto zu erstellen:

1. Öffnen Sie in Ihren Webbrowser https://azure.microsoft.com/free, und wählen Sie die Option, um den Einstieg mit einem kostenlosen Azure-Konto zu finden.
2. Melden Sie sich bei Ihrem Microsoft-Konto an, wenn Sie dazu aufgefordert werden. Wenn Sie ein Microsoft-Konto benötigen oder ein neues erstellen möchten, wählen Sie den Link „Create a New Microsoft Account“ (Neues Microsoft-Konto erstellen).
3. Sobald Sie sich bei einem Microsoft-Konto angemeldet haben, befolgen Sie die Aufforderungen, um ein kostenloses Azure-Konto zu erstellen (siehe Abbildung 1.2):
   - Geben Sie Ihre persönlichen Daten wie verlangt ein.
   - Minimieren Sie das Risiko von Missbrauch und Betrug, indem Sie eine Telefonnummer angeben, um Ihre Identität per SMS oder Anruf zu überprüfen.
   - Eine Kreditkarte ist ebenfalls für die Identitätsüberprüfung erforderlich, aber es gibt hier keinen Haken. Ihr Konto wird erst nach 30 Tagen belastet oder wenn Sie Ihre Gutschrift in Höhe von 200 US-Dollar aufgebraucht haben. Sie wechseln am Ende Ihres Testzeitraums nicht automatisch zu einem Abonnement mit nutzungsbasierter Zahlung. Es kann sein, dass Sie einen geringen Betrag von 1 US-Dollar (oder den Gegenwert in der Landeswährung) für die Überprüfung bezahlen müssen. Dieser wird nach ein paar Tagen zurückerstattet.
4. Überprüfen und akzeptieren Sie den Azure-Abonnement-Vertrag sowie die Datenschutzrichtlinien, und wählen Sie dann „Sign Up“ (Registrieren). Es kann einige Minuten dauern, bis Ihr Azure-Abonnement abgeschlossen wird.
5. Sobald die Registrierung abgeschlossen ist und das Azure Portal geladen wird, machen Sie eine schnelle Tour, um die Navigation zu erlernen.

Ihr Dashboard – die Startseite des Portals – sieht im Moment noch leer aus. Doch in Kapitel 2 tauchen Sie in die Erstellung Ihrer ersten VM ein, und Ihr Dashboard wird nach und nach wie in Abbildung 1.1 aussehen.

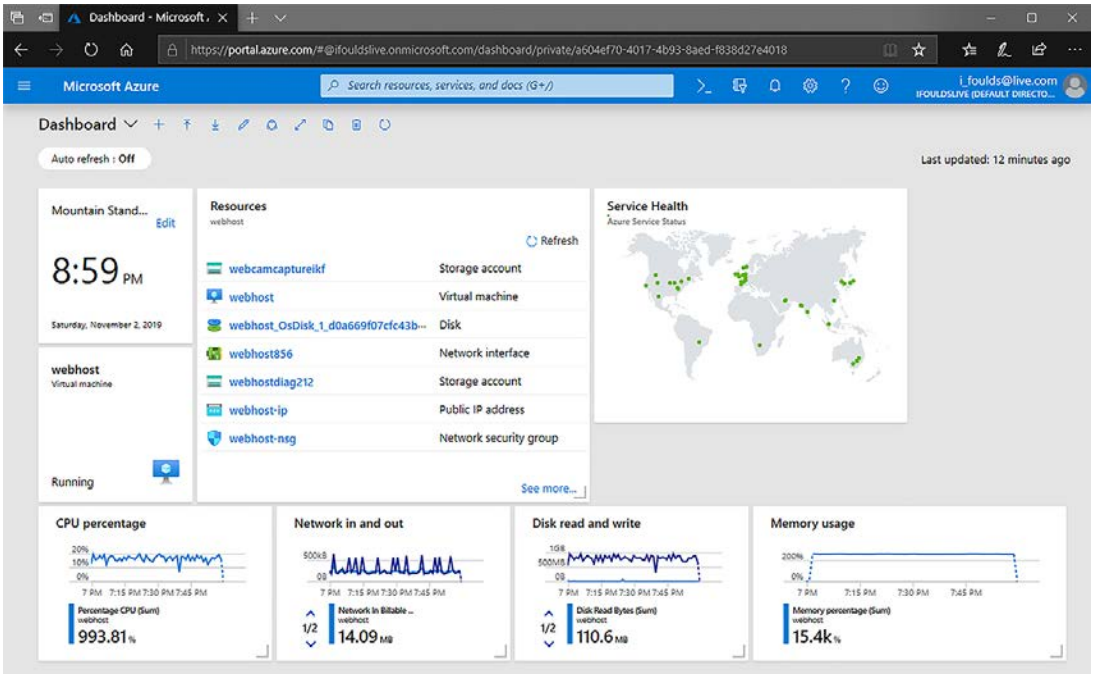


Abbildung 1.1 Das Azure Portal – bereit zur Erstellung Ihrer eigenen Anwendungen und Lösungen

Ist kostenlos wirklich kostenlos?

Azure hat einen Marketplace, der Hunderte von vorgefertigten Images (die Grundlage von VMs) und Lösungen enthält, die Sie bereitstellen können. In diesem Buch werden diese Marketplace-Angebote immer wieder genutzt – eine großartige Möglichkeit, schnell eine ganze Anwendungssuite bereitzustellen.

Nicht alle diese Azure Marketplace-Angebote sind kostenlos – einige Drittherausgeber kombinieren Lizenz- oder Supportkosten mit der von Ihnen bereitgestellten Lösung. Zum Beispiel kann für eine VM von Red Hat, die Sie bereitstellen, eine zusätzliche Gebühr für die Red Hat-Supportvereinbarung und -Lizenz anfallen. Diese Gebühren werden nicht durch Ihre Gutschrift im Rahmen der Testversion abgedeckt, sondern nur die grundlegende VM-Nutzung.

Für die Übungen in diesem Buch werden nur Ressourcen der kostenlosen Testversion verwendet. Wenn Sie sich auf die Suche nach anderen coolen Marketplace-Angeboten in Azure machen, achten Sie darauf, was Sie entwickeln. Jede Lösung, die mit zusätzlichen Gebühren verbunden ist, sollte dies klar und deutlich darlegen, bevor Sie sie nutzen.

1.3.2 Praktische Bonusübung: Erstellen eines kostenlosen GitHub-Kontos

GitHub ist ein kostenloser Webservice, den viele Organisationen und Einzelpersonen nutzen, um Projekte wie Code, Vorlagen und Dokumentation zu verwalten. Azure bietet Hunderte von kostenlosen Vorlagen und Skriptbeispielen, die Sie verwenden und zu denen Sie beitragen können. Die gegenseitige Unterstützung ist eine der Stärken der Open-Source-Community.

Für einige der Übungen in diesem Buch werden Ressourcen von GitHub verwendet. Sie benötigen dafür kein GitHub-Konto, doch wenn Sie kein Konto haben, können Sie Änderungen nicht speichern und nicht damit beginnen, Ihre eigene Sammlung von Vorlagen und Skripten zu erstellen. Die Erstellung eines GitHub-Kontos ist ein optionaler, aber sehr empfehlenswerter Teil des Aufbaus Ihrer Übungsumgebung:

1. Öffnen Sie in Ihrem Webbrowser https://github.com.
2. Geben Sie zum Erstellen eines kostenlosen GitHub-Kontos einen Benutzungsnamen, eine E-Mail-Adresse und ein Kennwort an. Sie erhalten eine Bestätigungsnachricht von GitHub.
3. Klicken Sie auf den Link in der Validierungs-E-Mail, um Ihr Konto zu aktivieren.
4. Sehen Sie sich einige Azure-Repositorys mit Beispielressourcen an:
   - Azure-Schnellstart-Vorlagen: https://github.com/Azure/azure-quickstart-templates
   - Azure CLI: https://github.com/Azure/azure-cli
   - Azure-DevOps-Dienstprogramme: https://github.com/Azure/azure-devops-utils
   - *Nutzen Sie Ihre Mittagspause, um mehr über Azure zu erfahren* – Buchressourcen: https://github.com/fouldsy/azure-mol-samples-2nd-ed

1.4 Eine kleine Hilfestellung

Dieses Buch kann nicht alle Möglichkeiten von Azure abdecken. Ich habe es zwar versucht, doch bis Sie dieses Kapitel fertiggelesen haben, gibt es sicherlich etwas Neues bei Azure. Cloud Computing entwickelt sich schnell und es werden immer wieder neue Dienste und Funktionen veröffentlicht. Ich mag ein wenig voreingenommen sein, doch wenn Sie beginnen, Azure zu erkunden, und mehr über zusätzliche Dienste erfahren möchten, ist die Website https://docs.microsoft.com/azure optimal für den Start geeignet. Jeder Azure-Dienst wird mit Schnellstart-Beispielen, Tutorials, Codebeispielen, Entwicklungsreferenzen und Architekturanleitungen dokumentiert. Sie können auch auf kostenlose und kostenpflichtige Support-Optionen zugreifen, wenn Sie Hilfe benötigen.

1.5 Verstehen der Azure-Plattform

Bevor Sie sich den Rest dieses Buchs vornehmen, lassen Sie uns einen Schritt zurückgehen und uns ein Verständnis dafür aneignen, was Azure ist und welche Dienste verfügbar sind. Wie bereits erwähnt, ist Azure ein Cloud Computing-Anbieter auf globaler Ebene. Zum Zeitpunkt des Verfassens dieses Buchs gab es 54 Azure-Regionen. Jede Region umfasst mindestens ein Rechenzentrum. Zum Vergleich: Die beiden anderen großen Cloud-Anbieter sind in 23 Regionen (Amazon Web Services [AWS]) und 20 Regionen (Google Cloud) vertreten.

Cloud Computing bietet mehr als nur Computingressourcen. Es gibt mehr als 100 Dienste in Azure, die in Familien verwandter Dienste wie Computing, Web und Mobile, Container und Identität zusammengefasst sind. Mit all diesen Diensten deckt Azure viele Servicemodelle ab. Versuchen wir bei einem Stück Pizza zu Mittag zu verstehen, was das bedeutet (Abbildung 1.2).

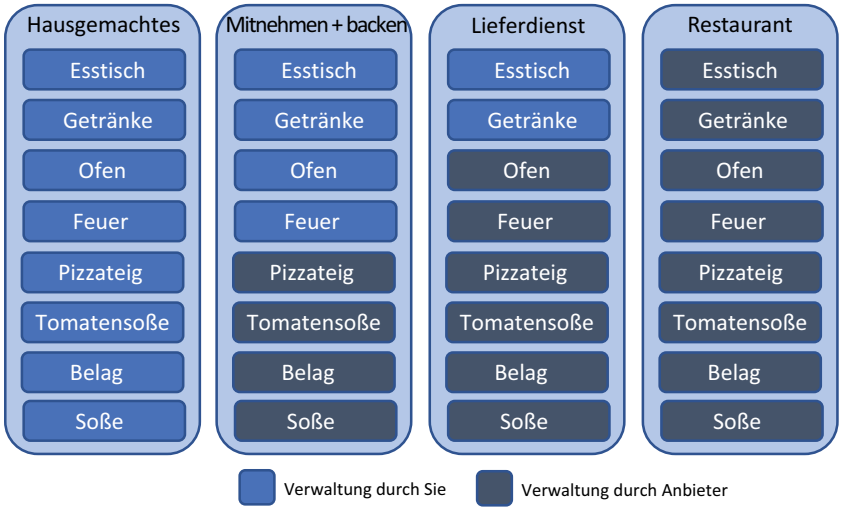


Abbildung 1.2 „Pizza-as-a-Service"-Modell. Beim Wechsel von hausgemachter Pizza, bei der Sie alles selbst zur Verfügung stellen, zum Restaurantmodell, bei dem Sie einfach nur ins Restaurant gehen, ändern sich die Verantwortlichkeiten und die Verwaltungsanforderungen entsprechend.

Beim „Pizza-as-a-Service"-Modell stehen Ihnen vier Möglichkeiten zur Auswahl. Während Sie die Modelle durchgehen, machen Sie sich immer weniger Gedanken um den eigentlichen Prozess, ein Stück Pizza zu essen:

- *Hausgemacht* – Sie machen den Teig, geben die Sauce, den Belag und Käse hinzu, backen die Pizza im Ofen, besorgen Getränke und setzen sich zum Essen an den Tisch.
- *Tiefkühlpizza* – Sie kaufen eine fertige Pizza. Sie müssen sie nur im Ofen backen, Getränke besorgen und sich zum Essen an den Tisch setzen.
- *Lieferdienst* – Sie bestellen eine Pizza, die zu Ihnen nach Hause geliefert wird. Sie müssen nur Getränke besorgen und sich zum Essen an den Tisch setzen.
- *Restaurant* – Sie gehen nach draußen, um mit minimalem Aufwand Pizza zu essen.

Da Sie hungrig sind, sehen wir uns das herkömmlichere Modell an, das einige Computingressourcen beinhaltet (Abbildung 1.3). Dieses Modell sieht schon eher aus wie etwas, das Sie in Azure sehen.

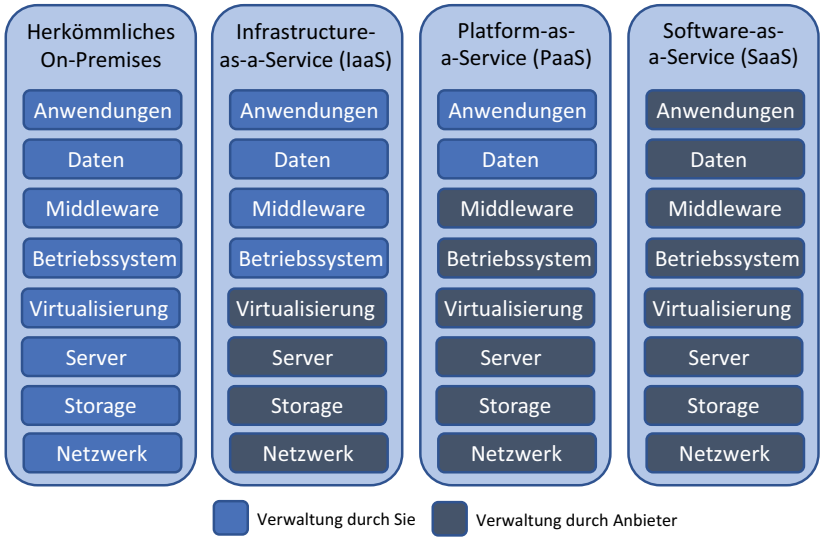


Abbildung 1.3 Cloud Computing-Servicemodell

Wenn Sie die Modelle durchgehen, verwalten Sie immer weniger der zugrunde liegenden Ressourcen und haben mehr Zeit und Energie für Ihre Kund*innen:

- *On-Premises* – Sie konfigurieren und verwalten das gesamte Rechenzentrum, z. B. Netzwerkkabel, Speicher und Server. Sie sind verantwortlich für alle Teile der Anwendungsumgebung, des Supports und der Redundanz. Mit diesem Ansatz haben Sie maximale Kontrolle, aber auch viel Verwaltungsaufwand.
- *Infrastructure-as-a-Service (IaaS)* – Sie kaufen die grundlegenden Computing-Ressourcen von einem Anbieter, der die Kerninfrastruktur verwaltet. Sie erstellen und verwalten die VMs, Daten und Anwendungen. Der Cloud-Anbieter ist für die physische Infrastruktur, die Hostverwaltung und die Resilienz verantwortlich.

Sie haben vielleicht weiterhin ein Infrastruktur-Team, das bei der Unterstützung und Bereitstellung von VMs hilft, aber Sie sparen Zeit und Kosten für die Verwaltung der physischen Arbeitsgeräte.

Dieser Ansatz ist gut geeignet, wenn Sie Anwendungen aus der On-Premises-Umgebung verlagern. Die Verwaltung und der Betrieb ähneln oft einer On-Premises-Umgebung. Aus diesem Grund können sich der Geschäftsbereich, die IT und die Anwendungseigentümer*innen mit IaaS auf natürliche Weise weiterentwickeln und sich mit der Cloud vertraut machen.

- *Platform-as-a-Service (PaaS)* – Sie kaufen den zugrunde liegenden Plattform-Stack von einem Anbieter, der das Betriebssystem und die Patches verwaltet, und verwenden Ihre Anwendungen und Daten. Machen Sie sich keine Gedanken um VMs oder das virtuelle Netzwerk. Ihr Betriebsteam kann sich so mehr Zeit für die Optimierung der Zuverlässigkeit und Leistung von Anwendungen nehmen.

  Mit diesem Ansatz können sich der IT- und Geschäftsbereich mit dem Ausführen von Anwendungen in der Cloud vertraut machen. Ihr Fokus liegt auf den Anwendungen und Ihren Kund*innen, und Sie müssen sich weniger Gedanken um die Infrastruktur machen, um diese Apps auszuführen.

- *Software-as-a-Service (SaaS)* – Sie benötigen nur Zugriff auf Software, und ein Anbieter stellt alles andere zur Verfügung. Entwickler*innen können mit einer bestehenden Plattform entwickeln, um Anpassungen oder einzigartige Funktionen zur Verfügung zu stellen, ohne eine große Codebasis pflegen zu müssen.

  Dieser Ansatz ist anfangs oft abschreckend, aber wahrscheinlich kennen und nutzen Sie bereits ein erfolgreiches SaaS-Angebot wie Salesforce, Office 365 oder die Google Mail oder Docs-Suite. Sie verwenden E-Mail-Programme, erstellen Dokumente bzw. Präsentationen oder verwalten Kundenkontaktinformationen und Vertriebsinformationen. Ihr Fokus liegt auf den Inhalten, die Sie erstellen und verwalten, und nicht darauf, wie die Anwendung ausgeführt werden kann.

Die meisten in Azure erstellten Dinge fallen in den IaaS- und PaaS-Bereich. Zu den hauptsächlichen Anwendungsfällen gehören VMs und virtuelle Netzwerke (IaaS) oder die Dienste Azure Web Apps, Functions und Cosmos DB (PaaS). Wenn Sie Entwickler*in sind, interessieren Sie die PaaS-Lösungen wahrscheinlich am meisten, denn Microsoft deckt die Infrastrukturteile ab, damit Sie sich auf Ihren Code konzentrieren können. IT-Profis konzentrieren sich wahrscheinlich mehr auf die IaaS-Lösungen, um die Azure-Infrastruktur zu entwickeln und zu steuern.

Man lernt nie aus

Vergessen Sie nicht, dass ein IT-Profi auch beim Wechsel eines Unternehmens vom IaaS- zum PaaS-Modell nicht überflüssig wird. Beim Entwerfen einer Lösung oder ihrer Problembehebung ist es wichtig zu verstehen, was hinter den Kulissen der PaaS-Ebene passiert. Wenn Sie ein IT-Profi sind, überspringen Sie die Kapitel bezüglich PaaS-Lösungen in Azure nicht – es erzeugt einiges an Mehrwert für Ihr Geschäft und Ihre Kund*innen, wenn Sie den Wechsel zu diesem Bereitstellungsmodell verstehen.

1.5.1 Virtualisierung in Azure

Virtualisierung ist die wahre Magie hinter Azure. Die IaaS-, PaaS- und SaaS-Modelle nutzen Virtualisierung, um die entsprechenden Dienste zu betreiben. Das Konzept der Virtualisierung ist nichts Neues und reicht bis in die Mainframe-Tage der 60er Jahre zurück. Mitte der 2000er Jahre nahm die Servervirtualisierung im Rechenzentrum Fahrtwind auf, und mittlerweile werden nur noch wenige Workloads auf Bare-Metal-Servern bereitgestellt, statt virtualisiert zu werden.

Ganze Bücher widmen sich der Virtualisierung, hier erhalten Sie aber eine schnelle Übersicht: Bei der Virtualisierung werden die physischen Ressourcen eines Servers logisch in virtuelle Ressourcen geteilt, auf die individuelle Workloads sicher zugreifen können. Eine VM ist eine der häufigsten Ressourcen im Cloud Computing. Eine VM enthält eine virtuelle CPU (vCPU), virtuellen Arbeitsspeicher (vRAM), virtuellen Speicher (vDisk) und virtuelle Netzwerkkonnektivität (vNIC), wie in Abbildung 1.4 gezeigt.

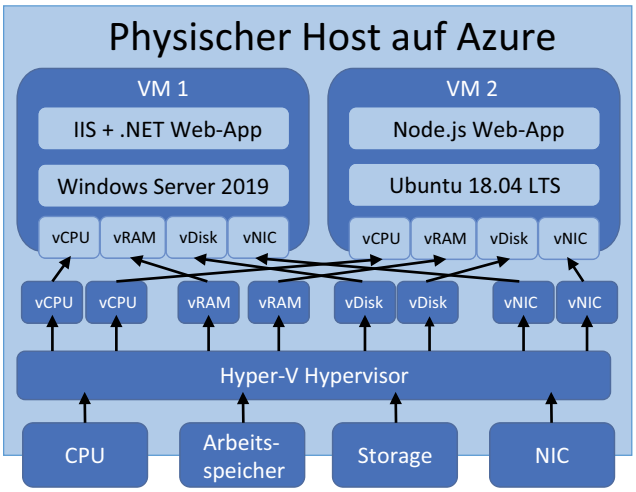


Abbildung 1.4 Virtualisierung in Aktion auf einem physischen Host in Azure

Neben physischen Servern werden auch Speicher und Vernetzung häufig virtualisiert, wodurch die Azure-Plattform schnell alles definieren kann, was Sie bei der Software benötigen. Es ist keine physische Interaktion oder manuelle Konfiguration der Geräte erforderlich. Sie müssen nicht warten, bis ein anderes Team eine IP-Adresse bereitstellt, einen Netzwerkport öffnet oder Speicherplatz für Sie hinzufügt.

Im Kern läuft Azure unter Windows – gewissermaßen. Eine modifizierte Version des Hyper-V-Hypervisors unterstützt die Computing-Server. Hyper-V ist ein Typ-1-Hypervisor (Bare-Metal), der bereits seit einem Jahrzehnt im Windows Server bereitsteht. Keine Sorge, Sie können Linux auch weiterhin als vollständig unterstützten, erstklassigen Workload ausführen. Microsoft leistet einen großen Beitrag zur Linux-Community und zum Linux-Kernel. Einige der wichtigsten Software-definierten Netzwerke in Azure werden mit einer maßgeschneiderten Lösung auf Basis von Debian Linux – Software for Open Networking in the Cloud (SONiC) – betrieben, die Microsoft als Open Source bereitstellt. Sie können hier einen virtuellen Rundgang durch die Rechenzentren von Microsoft unternehmen: https://azure.microsoft.com/global-infrastructure

1.5.2 Verwaltungstools

Wie können Sie die vielen Azure-Dienste nutzen? Ganz nach Ihrem Geschmack natürlich! Wenn Sie alles in einem Webbrowser auswählen möchten, gibt es dafür ein tolles webbasiertes Portal. Sie sind mit PowerShell vertraut? Wie Sie vermutlich bereits erwarten, gibt es ein Azure PowerShell-Modul. Es gibt auch eine plattformübergreifende Befehlszeilenschnittstelle (CLI), die ideal für macOS und Linux geeignet ist. Entwickler*innen können mit Azure außerdem über REST APIs interagieren, die eine Vielzahl von gängigen Sprachen nutzen wie .NET, Python und Node.js.

AZURE PORTAL

Das Azure Portal sollte mit jedem modernen Webbrowser funktionieren. Es ist eine praktische Art, Azure zu verwenden, ohne etwas auf Ihrem Computer zu installieren. Das Portal bietet auch eine großartige Möglichkeit, durch eine schnelle visuelle Darstellung aller Aspekte zu lernen, wie Ressourcen erstellt und verwaltet werden.

Azure werden ständig neue Funktionen und Dienste hinzugefügt, sodass sich das Portal immer wieder verändert. Daher weicht die tatsächliche Darstellung unter Umständen leicht von den Screenshots in diesem Buch oder der Online-Dokumentation und den Blogs ab. Die Formulierung einer Schaltfläche kann sich etwas ändern, oder es wird eine neue Option hinzugefügt, doch die zentralen Abläufe bleiben alle gleich. Willkommen in der schönen neuen Welt des Cloud Computing!

AZURE CLOUD SHELL

Für diejenigen, die unbedingt mit dem Eintippen von Befehlen beginnen möchten, enthält das Portal auch Azure Cloud Shell (siehe Abbildung 1.5). Diese Shell ist eine webbasierte interaktive Konsole, die eine Bash-Shell, Azure CLI und einige vorinstallierte Tools wie Git und Maven für die Anwendungsentwicklung zur Verfügung stellt. Es gibt auch eine PowerShell-Version der Cloud Shell, die, wie der Name schon sagt, Zugriff auf die neuesten Azure PowerShell-Cmdlets bietet.

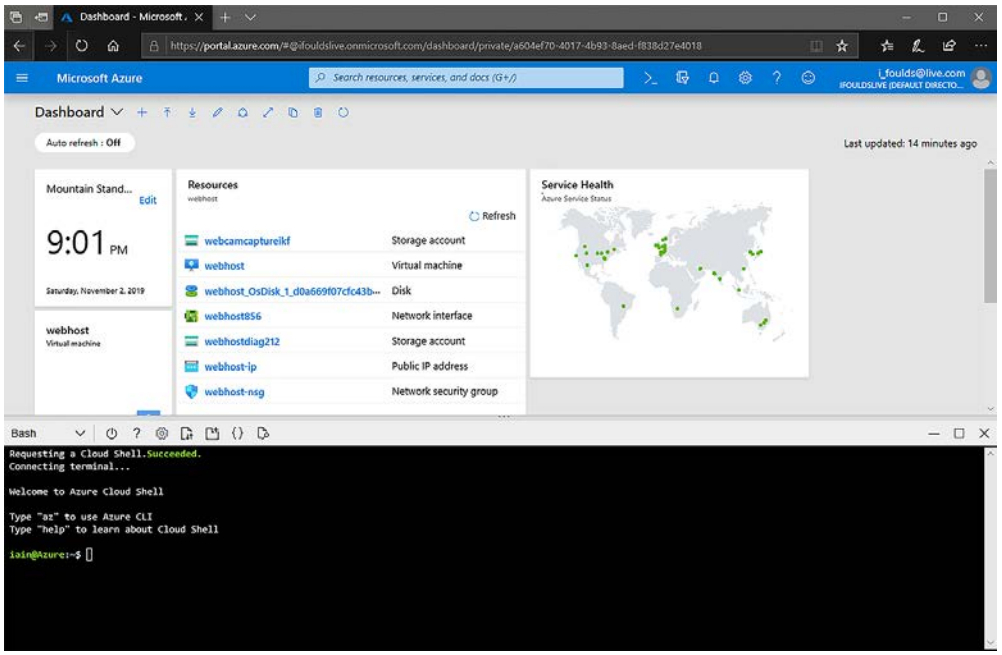


Abbildung 1.5 Azure Cloud Shell im webbasierten Portal

Sie können über einen Webbrowser von jedem Computer aus auf Azure Cloud Shell zugreifen, ohne Tools unter https://shell.azure.com installieren zu müssen. Editoren wie Visual Studio Code (https://code.visualstudio.com) ermöglichen Cloud Shell-Zugriff über die Anwendung. Es gibt sogar eine Azure-App für iOS und Android, mit der Sie die Azure Cloud Shell direkt über Ihr Smartphone nutzen können.

Mit Azure Cloud Shell haben Sie immer Zugriff auf die neueste Version der CLI oder der PowerShell-Tools. Mit dem persistenten Speicher können Sie Skripte, Vorlagen und Konfigurationsdateien erstellen und speichern.

Lokale Azure CLI- und PowerShell-Tools

Obwohl Azure Cloud Shell gewisse Vorteile hat, benötigen Sie damit oft Zugriff auf Ihr lokales Dateisystem und Ihre Tools. Sie können die Azure CLI oder Azure PowerShell lokal installieren, sodass Sie mit lokalen Ressourcen und Azure-Ressourcen arbeiten können.

In diesem Buch verwenden wir meist Azure CLI (genau genommen Azure CLI 2.0). Es mag seltsam erscheinen, dies der nativen Microsofts PowerShell vorzuziehen – der Vorteil dabei ist, dass Muster und Übungen sowohl mit Azure Cloud Shell als auch lokal auf Ihrem Computer funktionieren können, unabhängig davon, welches Betriebssystem Sie verwenden. Obwohl diese Informationen nicht zum Aufbau Ihrer praktischen Übungsumgebung gehören, wird in den folgenden Anleitungen detailliert beschrieben, wie Sie die Azure-Verwaltungstools auf Ihrem Computer installieren:

- *Erste Schritte mit Azure PowerShell:* https://docs.microsoft.com/powershell/azure/get-started-azureps
- *Installieren der Azure CLI*: https://docs.microsoft.com/cli/azure/install-azure-cli

2 Erstellen einer virtuellen Maschine

Möchten Sie erfahren, wie schnell Sie einen Webserver in Azure einrichten können? In diesem Kapitel befassen wir uns direkt mit der Entwicklung eines grundlegenden Webservers – einer der häufigsten Anfragen, wenn es um VMs geht: Dieser Workload ist ein großartiges Beispiel für die Kernkomponenten von Infrastructure-as-a-Service (IaaS) in Azure.

Nehmen wir an, Sie arbeiten für eine*n Pizzabäcker*in, der bzw. die seinen bzw. ihren Betrieb ausbauen und Onlinebestellungen für einen Pizzaliefer- oder -abholdienst annehmen möchte. Zum Aufbau einer Onlinepräsenz benötigen Sie eine Website. In den ersten Teilen dieses Buchs erforschen wir die verschiedenen Funktionen und Dienste in Azure, die es Ihnen ermöglichen, sowohl IaaS- als auch PaaS-Webanwendungen zu entwickeln und auszuführen. Sie können damit beginnen, fundierte Entscheidungen darüber zu treffen, wann eine VM entwickelt und ausgeführt werden soll, um eine Website zu betreiben, und wann Sie PaaS dazu verwenden könnten. Der erste Schritt besteht jedoch darin, einen Webserver zu entwickeln.

In diesem Kapitel erstellen Sie eine Ubuntu Linux-VM und installieren einen grundlegenden Webserver. Machen Sie sich keine Gedanken über die Verwendung von Linux – in der Übung am Ende des Kapitels erstellen Sie eine

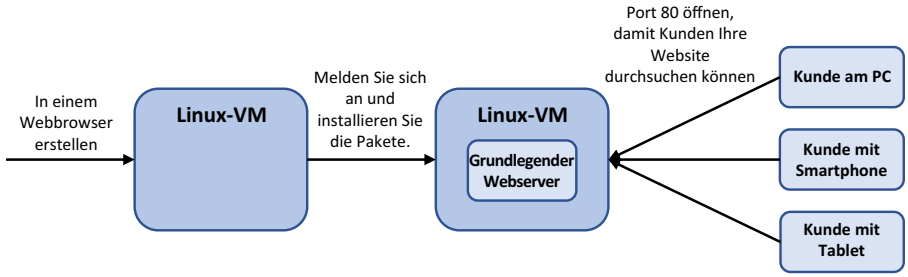


Abbildung 2.1 In diesem Kapitel erstellen Sie eine grundlegende VM, loggen sich ein, um einen Webserver zu installieren, und öffnen dann einen Netzwerkport, damit Kund*innen auf der Beispielwebsite surfen können.

Windows-VM. Ubuntu ist eine gängige Webserverplattform. Sie bietet eine großartige Möglichkeit, um mehr über SSH-Authentifizierung mit öffentlichen Schlüsseln zu lernen. Im Anschluss erfahren Sie, wie Sie einen Netzwerkport öffnen können, damit Kund*innen auf Ihre Website im Internet zugreifen können. Eine umfassende Übersicht über diese grundlegende Umgebung finden Sie in Abbildung 2.1.

2.1 Grundlagen zur Konfiguration virtueller Maschinen

VMs gehören zu den häufigsten Bausteinen, die Sie verwenden werden, wenn Sie Anwendungen in der Cloud ausführen möchten. Warum? Sie sind in der Regel vertrautes Gebiet. Die meisten IT-Abteilungen führen viele Workloads mit Hyper-V oder VMware in einer On-Premises-Umgebung aus, sodass Sie wahrscheinlich bereits Erfahrung mit der Erstellung und Ausführung von VMs haben. Unternehmen unternehmen ihre ersten Schritte in Azure oft mit VMs, da IaaS-Workloads nicht die große mentale Anpassung erfordern, die vorgenommen werden müsste, wenn Sie mit dem Ausführen von PaaS-Workloads beginnen.

Es gibt Lösungen für die Migration von VMs aus einer On-Premises-Umgebung wie Hyper-V oder VMware nach Azure. Bevor Sie sich allerdings zu sehr in die Möglichkeiten von Azure vertiefen (von denen wir einige in späteren Kapiteln erkunden werden), sollten Sie sich einige Grundlagen ansehen. Die folgenden Seiten enthalten möglicherweise bekannte Überlegungen und Optionen, die Sie auch bei On-Premises-VMs haben. Umso besser! Wenn dies Neuland für Sie ist – keine Sorge! Ein Großteil der Verwaltung ist in Azure abstrahiert, und Dinge wie virtuelle Netzwerke werden in der Regel einmal erstellt und konfiguriert und dann in Ruhe gelassen. In den folgenden Kapiteln werden wir die einzelnen Bereiche näher beleuchten, also atmen Sie tief durch, und unternehmen Sie einen Schritt nach dem anderen.

2.1.1 Regionen und Verfügbarkeitsoptionen

Azure ist in Regionen auf der ganzen Welt unterteilt, wobei jede Region über ein oder mehrere Rechenzentren verfügt. Diese Rechenzentren stellen die zentralen Rechen-, Speicher- und Netzwerkressourcen für die Ausführung Ihrer Anwendungen und Workloads bereit. Azure wird in mehr als 50 Regionen eingesetzt, wobei die Liste alle paar Monate weiter wächst. Bei so vielen Regionen können Sie Anwendungen problemlos in der Nähe Ihrer Mitarbeiter*innen oder Kunden*innen einsetzen. Diese regionale Verfügbarkeit verringert die Latenzzeiten und verbessert die User Experience der Anwender*innen.

Eine Azure-Region bietet möglicherweise nicht jeden einzelnen in Azure verfügbaren Dienst an. Bei Hunderten von verfügbaren Diensten sind die gängigsten Kerndienste in der Regel überall verfügbar, es werden aber in der Regel auch neue oder Nischendienste mit der Zeit eingeführt. Wenn Sie Ihre Anwendungen in Azure planen, informieren Sie sich unter https://azure.microsoft.com/global-infrastructure/services über die Produktverfügbarkeit nach Region.

In Kapitel 8 werden wir uns mit einigen der Hochverfügbarkeitsoptionen wie Verfügbarkeitsgruppen und Verfügbarkeitszonen beschäftigen. Mit diesen Redundanzoptionen kann Azure mehrere Instanzen Ihrer VMs oder Anwendungen innerhalb eines einzelnen Rechenzentrums oder über eine ganze Region hinweg verteilen. So können Sie Ihre Toleranz gegenüber Wartungsupdates oder Hardwareausfällen festlegen. In den ersten Kapiteln dieses Buches werden Sie in der Regel nur eine oder zwei VMs erstellen, sodass Sie sich über diese Verfügbarkeitsoptionen noch keine Gedanken machen müssen.

2.1.2 VM-Images

Um eine VM zu erstellen, benötigen Sie einen Ausgangspunkt. In der Regel hängt dieser Ausgangspunkt von der Wahl des Betriebssystems ab: Windows oder Linux. Als Nächstes folgt die Entscheidung, welche Windows-Version (z. B. Windows Server 2016 oder 2019) oder welche Linux-Distribution (z. B. Ubuntu, Red Hat Enterprise Linux oder SUSE) verwendet werden soll.

Ein *Image* – ein vorkonfiguriertes Betriebssystem-Bundle mit grundlegenden Konfigurationsoptionen – ist der Ausgangspunkt dafür. Azure enthält Hunderte dieser vorgefertigten Images im Azure Marketplace, die Sie bei der Erstellung von VMs verwenden können. Abhängig von Ihrem aktuellen Lizenzmodell können Sie oft bestehende Windows-Lizenzen übernehmen oder sich für zusätzlichen Support von Canonical entscheiden, um beispielsweise Ubuntu Linux oder Updates von Red Hat zu nutzen.

Um die Dinge so einfach und kurz zu halten, dass Sie diese Lektionen in einer Mittagspause durchgehen können, werden Sie diese vorgefertigten Images in Azure im gesamten Buch verwenden. In der realen Welt möchten Sie wahrscheinlich Anpassungen an Ihre geschäftlichen Bedürfnisse und Anforderungen vornehmen. Dazu erstellen Sie häufig ihre eigenen VM-Images. Der Workflow zum Erstellen und Verwalten der VMs ist derselbe wie bei den Azure Marketplace-Images, aber oft erfordert die Erstellung eigener Images umfassende Planung und anschließend stundenlanges Konfigurieren, Generalisieren und Erfassen der eigenen Images im Voraus.

Jetzt testen

Im Folgenden finden Sie einige Ideen, die Sie bei der Planung von Anwendungen in Azure berücksichtigen können. Sie klingen ziemlich banal, und oft trifft man diese Entscheidungen automatisch, ohne viel nachzudenken. Dennoch ist es wichtig, dass Sie Ihre Anwendungsanforderungen verstehen, bevor Sie mit der Erstellung und Ausführung von Anwendungen beginnen!

- In welchen Regionen soll Ihre Anwendung ausgeführt werden? Haben Sie eine große Konzentration von Benutzer*innen in einer bestimmten Region? Wie können Sie Redundanz bereitstellen?

  Wenn Sie interne Anwendungen erstellen, führen Sie diese in der Azure-Region aus, die Ihren Benutzer*innen am nächsten liegt. Wenn Sie zum Beispiel ein großes Büro in Houston, Texas, haben (vielleicht mögen Sie Raumfahrzeuge!), führen Sie Ihre Azure-Anwendungen und VMs im Süden der USA aus.

Wenn Sie externe Anwendungen entwickeln, rechnen Sie dann mit Kund*innen aus bestimmten Regionen? Diese Konfiguration kann mehrere Instanzen erfordern, die in verschiedenen Regionen eingesetzt werden (und auch hohe Verfügbarkeit bieten). Wir werden uns mit dieser Konfiguration in Kapitel 12 befassen.

- Müssen Sie eine Vielzahl von VM-Anpassungen bereitstellen? Wie lange dauert es, all diese Änderungen zu testen und zu validieren? Welche geschäftlichen Anforderungen treiben sie an?

  In einer traditionellen On-Premises-Umgebung wird oft viel Zeit damit verbracht, vorkonfigurierte Images für die Bereitstellung zu erstellen. In der Cloud sollten Sie versuchen, diese Zeit zu minimieren. Die vorgefertigten Azure-Images enthalten die neuesten Sicherheitsupdates. Sie werden für Sie getestet und dann geografisch repliziert, um eine schnelle Bereitstellung zu ermöglichen.

  Wenn Sie Ihre eigenen Images erstellen, verwenden Sie Funktionen wie Azure Shared Image Gallery, um diese Images nach Bedarf zu verteilen und zu replizieren (https://docs.microsoft.com/azure/virtual-machines/windows/shared-image-galleries).

2.1.3 VM-Größen

Es gibt verschiedene Familien von VM-Größen bei Azure. Diese Familien enthalten Gruppen ähnlicher virtueller Hardwaretypen, die auf bestimmte Workloads ausgerichtet sind. Die Größen werden manchmal aktualisiert, wenn neue Hardware- und Workload-Angebote zur Verfügung gestellt werden, doch die Kernfamilien bleiben unverändert. Folgende Familientypen sind vorhanden:

- *Allgemeine Verwendung* – gut geeignet für Entwicklung und Test, Produktionsdatenbanken mit geringer Nutzung oder Webserver
- *Computing-optimiert* – Hochleistungs-CPUs, zum Beispiel für Produktionsanwendungsserver
- *Speicheroptimiert* – größere Arbeitsspeicheroptionen, zum Beispiel bei Bedarf an der Ausführung großer Datenbanken oder Aufgaben, die viel In-Memory-Datenverarbeitung erfordern
- *Speicheroptimiert* – geringe Latenz, hohe Datenträgerleistung für datenträgerintensive Anwendungen
- *GPU* – NVIDIA-basierte grafikoptimierte VMs, bei Bedarf an Grafik-Rendering oder Videobearbeitung
- *Hochleistungs-Computing* – viel von allem: viel CPU, viel Speicher, viel Netzwerkdurchsatz für die anspruchsvollsten Workloads

Wie groß kann eine VM in Azure sein? Die Dinge verbessern sich ständig, aber zum Zeitpunkt des Verfassens dieses Buchs ist die größte VM, die Sie erstellen können, eine der Mv2-Serie (Teil der arbeitsspeicheroptimierten Familie) mit 208 virtuellen CPUs und 5,7 TiB Arbeitsspeicher. Das sollte für einen anständigen Minecraft-Server genügen, glauben Sie nicht?

Das Wichtigste, was Sie hier lernen sollten, ist, dass die Anzahl der VMs und die Anzahl der CPU und die Größe des Arbeitsspeichers, die Sie in Azure anfordern können, nur durch Ihr Budget begrenzt sind. In der herkömmlichen On-Premises-Welt wäre es wahrscheinlich schwierig, VMs dieser Größe zu erstellen.

Wenn Sie eine VM im Azure Portal oder über die Befehlszeilenschnittstelle oder PowerShell erstellen, müssen Sie die Größe der zu verwendenden VM auswählen. Eine gängige VM-Größe, wie z. B. D2s_v3, wird oft als Standard für den Einstieg verwendet. Das ist wahrscheinlich viel zu viel Leistung für einen einfachen Webserver, aber die Erstellung der VM und die Installation der benötigten Pakete ist ganz einfach.

Im Azure Portal können Sie nach einer ungefähren Größe (z. B. klein, mittel oder groß) oder einer bestimmten Familie (z. B. universelle oder speicheroptimierte VMs) filtern. Die geschätzten monatlichen Kosten werden ebenfalls angezeigt, damit Sie eine Vorstellung davon bekommen, wie teuer jede VM ist. Achten Sie auf die Kosten, denn sie können sich schnell summieren! In der Regel können Sie die VM-Größe ändern, nachdem die VM in Betrieb genommen wurde. Die VM muss allerdings heruntergefahren und neu gestartet werden, um den Vorgang abzuschließen.

VM-Kosteneinsparungen

Die standardmäßig erstellten VMs sind für Ihre Bedürfnisse wahrscheinlich zu leistungsstark, doch sie können schnell bereitgestellt und verwendet werden, was dazu beiträgt, dass Sie in Ihrer Mittagspause weniger Zeit mit der Installation von Paketen verbringen müssen.

Achten Sie in der Praxis auf die Arbeitsspeicher-, CPU- und Speicheranforderungen Ihrer VMs. Erstellen Sie entsprechend große VMs. Dieser Ansatz ist der gleiche wie in der On-Premises-Welt, in der Sie am Ende vielleicht über VMs verfügen, die viel mehr Arbeitsspeicher haben oder denen viel mehr virtuelle CPUs zugeordnet sind als nötig.

Es gibt auch eine besondere Art von VM in Azure: die *B-Serie*. Diese VM-Größen verwenden Burstable-CPU- und -Speicherressourcen, und Sie können „Guthaben" für ungenutzte Computingressourcen ansparen. Wenn Sie Ihr Azure-Guthaben sparen möchten, können Sie diese Serie von VMs für die Übungen im Buch wählen. Sie haben einen niedrigeren Preis und sind ideal für Szenarien geeignet, in denen nicht immer viele CPU- und Arbeitsspeicherressourcen benötigt werden. Aber Vorsicht: Je nach Größe der von Ihnen erstellten VM der B-Serie verfügt sie möglicherweise über weniger CPU und Arbeitsspeicher als z. B. die D2s_v3-Serie, sodass sie etwas langsamer laufen wird.

2.1.4 Azure Storage

Der Storage für VMs ist bei Azure eine unkomplizierte Angelegenheit. Wie viele Datenträger möchten Sie verwenden, wie groß und von welchem Typ sollen sie sein? Die ersten beiden sind eigentlich nicht Azure-spezifisch, also lassen wir sie weg. Folgende Storage-Typen sind verfügbar:

- *Premium-SSD-Datenträger (Solid State Drive)* – verwenden Sie SSDs mit geringer Latenz und hoher Leistung. Diese eignen sich perfekt für Produktions-Workloads. Sie sollten hauptsächlich diesen Typ verwenden, um die beste Leistung Ihrer Anwendungen zu erhalten.

- *Standard-SSDs* – Datenträger: Verwenden Sie Standard-SSDs und bieten Sie eine konstante Leistung im Vergleich zu Festplatten (HDDs). Dieser Typ eignet sich hervorragend für die Entwicklung und Tests von Workloads oder für die Produktion mit knappem Budget und geringer Nachfrage, wie z. B. Webserver.
- *Standard-HDDs* – verwenden Sie herkömmliche rotierende Festplatten, die ideal für einen unregelmäßigen Datenzugriff wie für Archive oder Sicherungen sind. Dieser Typ ist nicht empfohlen für den Betrieb von Anwendungsworkloads.

Sie müssen nicht viel tiefer in die Besonderheiten des Speichers eintauchen, um schnell einen Webserver zu erstellen. In Kapitel 4 erfahren Sie mehr darüber, auch über Ultra-Disks, die nur für angeschlossene Datenträger gedacht sind. Im Moment reicht es aus, wenn Sie wissen, dass die Wahl einer VM-Größe bei der Definition der von Ihnen verwendeten Storage-Art helfen kann.

Die von Ihnen verwendeten virtuellen Datenträger werden unabhängig von ihrem Typ als *verwaltete Azure-Datenträger* bezeichnet. Diese verwalteten Datenträger ermöglichen es Ihnen, eine VM zu erstellen und zusätzliche Datenträger anzuhängen, ohne sich um die zugrunde liegenden Storage-Konten, Ressourcenbegrenzungen oder Leistungsquoten kümmern zu müssen. Verwaltete Datenträger werden im Ruhezustand außerdem automatisch verschlüsselt – Sie müssen zur Sicherung Ihrer Daten nichts konfigurieren. All dies und noch mehr wird in Kapitel 4 behandelt. Im Moment können Sie Azure in der Regel den am besten geeigneten Datenträger auf der Grundlage der von Ihnen gewählten VM-Größe erstellen lassen.

Jetzt testen

Um Ihr Wissen zu überprüfen, beantworten Sie bitte die folgenden Fragen:

- Welche Art von Datenträger bietet für die meisten Produktions-Workloads die beste Leistung?

  Für Produktionsworkloads sollten Sie in der Regel eine hochwertige SSD-Datenträger verwenden. Dieser Typ ist oft die Standardwahl, wenn Sie eine VM erstellen. Standard-SSD-Datenträger sind eine gute zweite Wahl, und Ultra-SSDs sollten nur für sehr datenträgerintensive Anwendungen verwendet werden, die eine geringe Latenzzeit erfordern. Obwohl die Kosteneinsparungen bei Standard-HDDs nur gering sind, ist die Leistung oft sehr gut, genau wie bei virtuellen On-Premises-Umgebungen.
- Welche VM-Familie ist eine gute Wahl für einen Datenbankserver?

  Eine speicheroptimierte VM ist eine gute Wahl, da Datenbanken oft mehr Speicher als CPU-Ressourcen benötigen. Versuchen Sie immer, den Ressourcenbedarf abzuschätzen und die Leistung nach der Bereitstellung zu überwachen. Scheuen Sie sich nicht, zu einer anderen VM-Größe zu wechseln, um die gewünschte Leistung zu erzielen.

2.1.5 Virtuelle Netzwerke

Es scheint offensichtlich, doch Ihre VM benötigt eine Netzwerkverbindung, wenn Sie möchten, dass jemand auf Ihre Anwendungen zugreifen kann. Für einen grundlegenden Webserver benötigen Sie sowohl ein virtuelles Netzwerk als auch externe Konnektivität. In Kapitel 5 wird der Kern des Azure-Netzwerks ausführlich behandelt, und in Kapitel 9 geht es um die Verteilung des Datenverkehrs auf mehrere VMs mithilfe von Load Balancers. Richtig cool wird es in Kapitel 11 mit Azure DNS und dem globalen Routing von Anwender*innen mit Traffic Manager. Ich werde Sie nicht zu einem oder einer Netzwerkingenieur*in machen, aber Sie werden in diesem Buch eine Menge über Azure-Netzwerke lernen!

Um mit den für dieses Kapitel benötigten Grundlagen zu beginnen, besteht ein virtuelles Netzwerk in Azure aus denselben Kernfunktionen wie ein normales physisches Netzwerk:

- Ein Adressraum und eine Subnetz-Maske, z. B. 10.0.0.0/16
- Mindestens ein Subnetz, mit dem Sie zum Beispiel externen, Datenbank- oder Anwendungsverkehr aufteilen können
- Virtuelle Netzwerkschnittstellenkarten (NICs), die VMs mit einem bestimmten Subnetz verbinden
- Virtuelle IP-Adressen, die Ressourcen wie einer virtuellen NIC oder einem Load Balancer zugeordnet sind

Sie können eine VM erstellen, die nur an ein virtuelles Netzwerk angeschlossen ist, ohne externe Konnektivität zu bieten, was für Backend-Datenbanken oder Anwendungsserver angebracht sein kann. Um sich für die Verwaltung und Wartung mit diesen VMs zu verbinden, können Sie eine virtuelle private Netzwerkverbindung (VPN) verwenden oder eine private, spezielle Verbindung zu Azure von Ihren On-Premises-Netzwerkgeräten. Bei Azure wird diese spezielle Verbindung *ExpressRoute* genannt.

Der grundlegende Webserver, den Sie in diesem Kapitel entwickeln werden, erfordert einen bestimmten Typ von virtueller IP-Adresse: eine öffentliche IP-Adresse. Diese öffentliche IP-Adresse ist der virtuellen NIC zugeordnet und ermöglicht es dem externen Datenverkehr, Ihre VM zu erreichen. Sie können dann den Datenverkehrsfluss zu Ihrer VM mit NSGs (Netzwerksicherheitsgruppen) steuern. Stellen Sie sich eine reguläre Firewall vor, die Sie verwenden, um verschiedene Ports und Protokolle zu öffnen oder zu schließen. In Azure schränken Netzwerksicherheitsgruppen den Datenverkehr standardmäßig ein und lassen nur den spezifischen Datenverkehr zu, den Sie definieren. Üblicher Datenverkehr, der zugelassen werden sollte, ist HTTP oder HTTPS auf den TCP-Ports 80 und 443. Die Remote-Verwaltung über das Remote Desktop-Protokoll (RDP) oder Secure Shell (SSH) kann ebenfalls geöffnet werden, wenn Sie vorsichtig sind. Dies werden Sie später in diesem Kapitel tun, um zu sehen, wie Sie eine Verbindung herstellen und einige Pakete installieren.

2.2 Erstellen eines SSH-Schlüsselpaars für die Authentifizierung

In der Übung am Ende des Kapitels erstellen Sie etwas, mit dem Sie wahrscheinlich schon vertraut sind: eine Windows Server-VM. Dieser VM-Typ verwendet eine kennwortbasierte Authentifizierung. Viele Anwendungen in der Cloud laufen unter Linux. Tatsächlich werden mehr als die Hälfte der VMs in Azure unter Linux ausgeführt. Unter Linux verwenden Sie normalerweise keine kennwortbasierte Authentifizierung, sondern SSH und ein öffentliches Schlüsselpaar. Um Ihre Kenntnisse zu erweitern, läuft der grundlegende Webserver in diesem Kapitel unter Linux, Sie müssen sich also mit der Erstellung und Verwendung von SSH vertraut

machen. Sie brauchen *keine* Linux-Erfahrung, um in der Cloud zu arbeiten, aber ich empfehle Ihnen dringend, einige der Grundlagen zu lernen!

SSH-Schlüsselpaare

SSH ist ein Protokoll, das für die sichere Kommunikation mit Remotecomputern verwendet wird. Es ist die am häufigsten genutzte Art, um sich bei Linux-VMs anzumelden. Es ist ähnlich wie die Verwendung einer RDP-Verbindung zu einer Windows-VM, außer dass in Linux die gesamte SSH-Sitzung in der Regel konsolenbasiert. Bei der Kryptografie mit öffentlichem Schlüssel können Sie für die Authentifizierung bei einer Linux-Remote-VM ein digitales Schlüsselpaar verwenden.

Ein SSH-Schlüsselpaar besteht aus zwei Teilen: einem öffentlichen und einem privaten Schlüssel. Der öffentliche Schlüssel wird auf Ihrer Linux-VM auf Azure gespeichert. Den privaten Schlüssel bewahren Sie als Kopie auf. Wenn Sie sich in Ihrer Linux-VM anmelden müssen, wird der öffentliche Schlüssel auf der Remote-VM mit dem privaten Schlüssel abgestimmt, den Sie lokal verwenden. Stimmen die Schlüsselpaare überein, werden Sie bei der VM angemeldet. Es steckt ein wenig mehr dahinter, aber im Großen und Ganzen ist die Kryptografie mit öffentlichem Schlüssel eine großartige Art und Weise, um Identitäten zu überprüfen.

Ich möchte, dass Sie sich daran gewöhnen, SSH-Schlüssel zu verwenden, um sich bei Linux-VMs anzumelden. SSH-Schlüssel sind viel sicherer als Kennwörter, da sie unter anderem nicht anfällig für Brute-Force-Kennwortangriffe sind. Vor allem in der Cloud sollten Sie sich immer auf die Sicherheit als zentrales Konzept konzentrieren.

Jetzt testen

Erstellen wir ein öffentliches SSH-Schlüsselpaar mit der Azure Cloud Shell:

1. Öffnen Sie in Ihrem Webbrowser https://portal.azure.com. Melden Sie sich bei dem Azure-Konto an, das Sie in Kapitel 1 erstellt haben, und wählen Sie dann das Cloud Shell-Symbol oben im Dashboard, wie in Abbildung 2.2 dargestellt. Sie können Cloud Shell auch direkt unter https://shell.azure.com öffnen.

Abbildung 2.2 Wählen Sie die Cloud Shell im Azure Portal aus, und starten Sie sie, indem Sie das Shell-Symbol auswählen.

2. Beim ersten Öffnen der Cloud Shell wird es einige Augenblicke dauern, da ein persistenter Speicher erstellt wird, der immer mit Ihren Sessions verbunden ist. Dieser Speicher ermöglicht es Ihnen, Skripte, Konfigurationsdateien und SSH-Schlüsselpaare zu speichern und abzurufen. Akzeptieren Sie alle Aufforderungen zum Erstellen dieses Speichers.
3. Wählen Sie gegebenenfalls Bash aus dem Dropdown-Menü in der oberen linken Ecke der Cloud Shell. PowerShell-Unterstützung ist ebenfalls verfügbar, obwohl wir uns in diesem Buch hauptsächlich auf Bash und die Azure CLI konzentrieren werden.
4. Geben Sie zum Erstellen eines Schlüsselpaars folgenden Befehl ein:

```
ssh-keygen
```

5 Akzeptieren Sie die Standardaufforderungen, indem Sie die EINGABETASTE drücken. Innerhalb weniger Sekunden erhalten Sie ein öffentliches SSH-Schlüsselpaar, das Sie mit all Ihren VMs verwenden können. Der `ssh-keygen`-Befehl wird auf einen 2.048-Bit-Längenschlüssel standardisiert und verwendet das RSA-Protokoll der Version 2. Dadurch entsteht eine ausgewogene Sicherheitsmaßnahme – der empfohlene Typ für die meisten Anwendungsfälle. Abbildung 2.3 zeigt ein Beispiel für ein vollständiges SSH-Schlüsselpaar in der Cloud Shell.

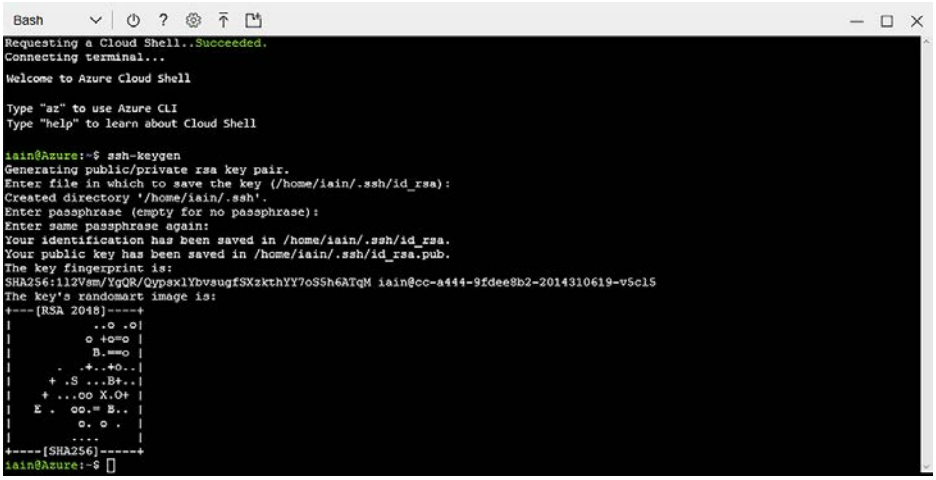


Abbildung 2.3 Ein SSH-Schlüsselpaar, das in der Azure Cloud Shell mit dem `ssh-keygen`-Befehl erstellt wurde

6 Geben Sie zur Anzeige Ihres öffentlichen Schlüssels und seiner Verwendung mit einer VM folgenden Befehl ein:

```
cat .ssh/id_rsa.pub
```

7 Wählen Sie die Ausgabe aus, und kopieren Sie sie in eine einfache Textdatei auf Ihrem Computer. Sie verwenden diesen öffentlichen Schlüssel, um in Abschnitt 2.3 eine VM zu erstellen. Auf diese VM wird im weiteren Verlauf des Buches von der Azure CLI aus verwiesen. Normalerweise müssen Sie nicht jedes Mal den gesamten Schlüssel kopieren und einfügen, aber es ist gut, wenn Sie zunächst einmal sehen, was passiert. Diese Informationen sind kein großes Geheimnis, es ist also völlig in Ordnung, wenn Sie mit Notepad oder TextEdit eine Kopie des Schlüssels erstellen und speichern. Gehen Sie beim Kopieren der Ausgabe des öffentlichen Schlüssels sorgfältig vor, da der Schlüssel auf zusätzliche Leerzeichen oder fehlende Zeichen empfindlich reagiert. Ein Beispiel für einen vollständigen öffentlichen SSH-Schlüssel ist:

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDPGaOBsfhJJOHAWAv+RLLR/vdUTzS9HOIj
➥JyzWWLsnu0ESH2M6R+YYPPNXv9X7dmVyM1zCXXEaLucpnyFjevbwPedxTgifyxgCFTgylr1
➥kg7o4EyCTGBGhTA+hSHuhXGXa12KPdKWehsPwHMa6Hs8fbt/in9Z1k2ZAwvbT+LWPcmJgNO
➥FuolIHOsOEeoQQqdXLrGa7NU/3fzSXdT9Y2BT1KLINc4KnwdOuONddLw3iANvK+Gkwax8iK
➥7IicKMoammwvJUCRf+MTEK9pZ84tfsc9qOIAdhrCCLbQhtoWjZpIwYnFk+SNBE8bZZtB8b2
➥vkDFNZlA5jcAd6pUR3tPuL0D iain@cc-a444-9fdee8b2-2014310619-v5cl5
```

TIPP Die Cloud Shell ist browserbasiert, sodass die Tastaturkürzel zum Kopieren und Einfügen ein wenig anders funktionieren können, als Sie es gewohnt sind. Mit Strg + Einfg und Umschalt + Einfg sollten Sie kopieren und einfügen können, statt wie normalerweise mit Strg + C und Strg + V.

2.3 Erstellen einer VM über Ihren Webbrowser

Nachdem Sie nun ein wenig über die Theorie der Azure-VMs erfahren und ein SSH-Schlüsselpaar erstellt haben, können Sie direkt loslegen und eine VM erstellen. Ich werde Ihnen den Einstieg erleichtern und Sie dann die VM auf der Grundlage des soeben Gelernten konfigurieren lassen. Passen Sie also unbedingt gut auf!

Die Azure CLI- und Azure PowerShell-Tools sind unglaublich leistungsfähig. Eine große Stärke von Azure ist jedoch das großartige Portalerlebnis, der mit viel Aufwand und Mühe entwickelt wurde. Das Azure Portal ist ein webbasiertes grafisches Tool, mit dem Sie sehen können, wie alle verschiedenen Komponenten zusammenkommen. Des Weiteren zeigt eine schnelle visuelle Überprüfung, ob alles in Ordnung ist. Das Portal enthält einige einzigartige Aspekte, die andere Tools nicht bieten. Es ist auch schnell einsatzbereit, da Sie nichts installieren müssen.

Jetzt testen

Bei der Erstellung einer VM in Azure stehen Ihnen viele Standardeinstellungen zur Verfügung, die Sie verwenden können, um die Anzahl der zu treffenden Auswahlen zu reduzieren. Sehen Sie sich für diese Übung die Ressourcen an, die Azure auf der Grundlage dessen, was Sie in Abschnitt 2.2 für Dinge wie Netzwerk und Speicher gelernt haben, erstellen wird:

1. Wählen Sie im Azure Portal (https://portal.azure.com) oben links im Dashboard „Create a Resource“ (Eine Ressource erstellen) aus. Beliebte Ressourcen sollten aufgelistet werden, einschließlich der neuesten Long Term Support(LTS)-Version von Ubuntu (zum Zeitpunkt der Erstellung dieses Artikels ist es Ubuntu Server 18.04 LTS).
2. Wählen Sie die LTS-Version aus.

   Sie können auch den Marketplace am oberen Rand des Fensters durchsuchen oder die Liste der übergeordneten Dienste (z. B. Compute und Netzwerk) durchsuchen, um eine Vorstellung davon zu bekommen, was für Ihren eigenen zukünftigen Bedarf noch verfügbar ist. Versuchen Sie, bei Ubuntu Server 18.04 LTS zu bleiben, damit Sie eine der kommenden Übungen zur Installation der Webserver-Komponenten durchführen können.
3. Erstellen Sie eine Ressourcengruppe für Ihren Webserver.

   Wenn Sie Ressourcen in Azure erstellen, sind sie logischerweise in einer Ressourcengruppe enthalten, die Sie definieren. Diese Gruppen enthalten in der Regel vergleichbare Ressourcen für Ihre Anwendungen. Kapitel 7 befasst sich mit der Planung und Verwaltung von Anwendungen mithilfe von Ressourcengruppen.

   Ich schlage fürs Erste vor, die Ressourcengruppen nach Kapiteln zu benennen, um den Überblick zu behalten. Nennen Sie die Ressourcengruppe in dieser Übung zum Beispiel `azuremol-chapter2`.
4. Geben Sie Ihrer VM einen Namen, z. B. webvm, und wählen Sie dann eine Region in Ihrer Nähe. Machen Sie sich vorerst keine Gedanken über die Redundanz der Infrastruktur.

   Sehen Sie sich die Optionen für das VM-Image an, nur um ein Gefühl für die anderen Optionen zu bekommen. Für diese Übung bleiben Sie aber bei Ubuntu

Server 18.04 LTS. Die Standard-VM-Größe ist für diese Übung gut geeignet, aber auch hier sollten Sie nachsehen, was verfügbar ist und wie Sie verschiedene Größen abfragen und mit welcher Hardware sie laufen. Sehen Sie, wie die Größen mit den VM-Familien übereinstimmen, die Sie zuvor in diesem Kapitel betrachtet haben.

5. Vergewissern Sie sich, dass Sie die SSH-Authentifizierung mit öffentlichem Schlüssel verwenden, und geben Sie dann einen Benutzungsnamen an, z. B. azuremol. Sie können diesen Benutzungsnamen verwenden, um sich in der nächsten Übung bei der VM anzumelden.
6. Kopieren Sie den öffentlichen SSH-Schlüssel, den Sie im vorherigen Abschnitt erstellt haben, und fügen Sie ihn ein. Achten Sie auch hier darauf, dass beim Kopieren und Einfügen des öffentlichen Schlüssels keine zusätzlichen Leerzeichen oder Formatierungen gesetzt werden. Der SSH-Schlüssel muss sich in einer Linie befinden. Selbst der Zeilenumbruch in Notepad kann Probleme verursachen! Das Azure Portal validiert den Schlüssel, bevor Sie fortfahren können.
7. Um sich in der nächsten Übung mit der VM zu verbinden und die Webserver-Komponenten zu installieren, öffnen Sie SSH auf Port 22.

   Das Öffnen von SSH auf einer öffentlichen VM ist keine gute Sicherheitspraxis. Kapitel 16 befasst sich mit der automatischen Öffnung und Einschränkung des Zugriffs durch Just-in-Time-VM-Zugriff.

   Sehen Sie sich einige der anderen Ports an, die Sie hier öffnen können. HTTP und HTTPS sind übliche Ports, die geöffnet werden müssen, und Sie sollen in diesem Kapitel einen Webserver erstellen, stimmt's? Schummeln Sie nicht und öffnen Sie diese Ports noch nicht. Ich möchte Sie in der nächsten Übung in die Azure CLI einführen, in der Sie HTTP-Datenverkehr zulassen.

Sichere Verbindung mit einem Bastion-Host

In einer Life-Umgebung dürfen Sie keine Remoteverwaltungsports für SSH oder RDP für das öffentliche Internet öffnen. Lassen Sie das wirklich bleiben! Befolgen Sie die bewährten Methoden, die Sie auch in der nicht cloudbasierten, On-Premises-Welt anwenden sollten, z. B. eine Verbindung nur bei Bedarf herzustellen und den Remote-Zugriff auf eine bestimmte Gruppe von Verwaltungsadressen zu beschränken.

Eine gängige Methode für den Remote-Zugriff ist die Verwendung eines Bastion-Hosts oder einer Jumpbox. Bei dieser Art der Konfiguration stellen Sie von Ihrem Laptop oder Desktop aus keine direkte Verbindung zu den Anwendungsservern her. Stattdessen stellen Sie eine Verbindung zu einem speziellen Bastion-Host her und verbinden sich dann mit dem Server, den Sie verwalten möchten. Bei diesem Ansatz wird der Zugriff auf eine begrenzte Anzahl von Adressen gesperrt, und eine sichere Remote-Verwaltung ist möglich.

Azure Bastion (https://docs.microsoft.com/azure/bastion) bietet einen verwalteten Ansatz für diesen Bedarf an sicheren Remote-Verbindungen. Sie erstellen einen Azure Bastion-Host in einem dedizierten Subnetz und verwenden diesen Host, um sich mit den VMs zu verbinden, auf denen Ihre Anwendungen laufen. Diese VMs müssen nicht öffentlich zugänglich sein. Sie können alles über das Azure Portal erledigen, ohne Netzwerkports für SSH oder RDP zu öffnen. Der Bastion-Host selbst wird in Bezug auf Sicherheitsupdates und Netzwerksicherheitsgruppenregeln für Sie verwaltet.

8. Sehen Sie sich einige der anderen VM-Konfigurationsoptionen für Speicher und Netzwerk an, um sich mit den Optionen vertraut zu machen. Sie können aber vorerst alles als Standard belassen.
9. Es gibt auch einige coole Verwaltungsoptionen, wie z. B. die Aktivierung der Option zur automatischen Abschaltung, von Sicherungen und Diagnosen, die in

den Kapiteln 12 und 13 behandelt werden. Deaktivieren Sie vorerst Dinge wie Startdiagnose und Diagnose des Gastbetriebssystems, da Sie ein Storage-Konto erstellen und konfigurieren müssen, damit diese funktionieren.

10 Wenn Sie alles vorbereitet haben, überprüfen und erstellen Sie Ihre Basis-VM.

2.4 Verbindungsherstellung zur VM und Installation des Webservers

Sobald Ihre VM in Betrieb ist, können Sie die SSH-Schaltfläche verwenden, die Sie vor einigen Minuten erstellt haben, um sich bei Ihrer VM anzumelden. Dann können Sie damit beginnen, den Webserver zu installieren und zu konfigurieren – alles über die Cloud Shell.

2.4.1 Verbindung zur VM mit SSH

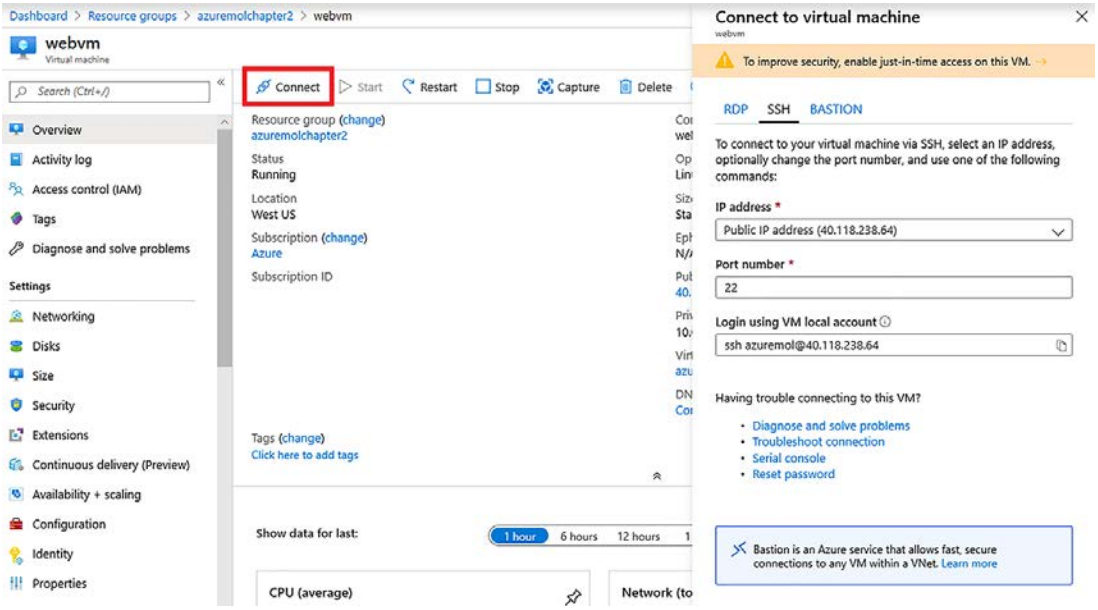
In diesem Abschnitt sehen wir uns an, wie Sie die Verbindungsdetails für Ihre VM schnell erhalten können.

Jetzt testen

Wenn Linux neu für Sie ist – keine Sorge! Führen Sie die nächsten Schritte aus, um sich bei Ihrer VM anzumelden:

1 Im Azure Portal können Sie virtuelle Maschinen über die Navigationsleiste auf der linken Seite des Bildschirms durchsuchen und auswählen. Es dauert ein paar Minuten, bis die VM aus der vorangegangenen Übung erstellt ist. Wählen Sie daher die Schaltfläche „Refresh“ (Aktualisieren), bis der VM-Status *Running* (Wird ausgeführt) zeigt. Wenn Sie bereit sind, wählen Sie Ihre VM und dann „Connect“ (Verbinden), wie in Abbildung 2.4 dargestellt.

Abbildung 2.4 Wählen Sie Ihre VM im Azure Portal aus und wählen Sie dann „Connect“ (Verbinden), um die SSH-Verbindungsinformationen zu generieren.



Bei einer Linux-VM wird Ihnen der SSH-Befehl angezeigt, der Ihren Namen und die öffentliche IP-Adresse enthält. Kopieren Sie diesen SSH-Verbindungsbefehl, z. B. `ssh azuremol@104.209.208.158`.

Bei einer Windows-VM laden Sie über die Schaltfläche „Connect“ (Verbinden) eine RDP-Verbindungsdatei auf Ihren Computer herunter, die bereits die öffentliche IP-Adresse Ihrer VM enthält.

2 Öffnen Sie bei Bedarf die Cloud Shell erneut. Wenn Sie voraussichtlich zwischen Cloud Shell und dem Portal umschalten möchten, können Sie die Cloud Shell minimieren, damit sie im Hintergrund verfügbar bleibt.

3 Fügen Sie den SSH-Befehl in die Cloud Shell ein, und drücken Sie die EINGABETASTE. Der SSH-Schlüssel, den Sie zuvor erstellt haben, wird automatisch zur Authentifizierung verwendet.

Bei der ersten Verbindung zu einer VM mit SSH werden Sie dazu aufgefordert, die VM in eine Liste vertrauenswürdiger Hosts einzufügen. Dies ist eine weitere Sicherheitsebene, die SSH bietet. Wenn jemand versucht, den Datenverkehr abzufangen und Sie zu einer anderen Remote-VM zu leiten, erkennt Ihr lokaler SSH-Client, dass sich etwas geändert hat, und warnt Sie vor der Verbindung.

Akzeptieren Sie die Aufforderung, um die Remote-VM-Verbindung zu speichern. Abbildung 2.5 zeigt den SSH-Verbindungsprozess in Azure Cloud Shell.

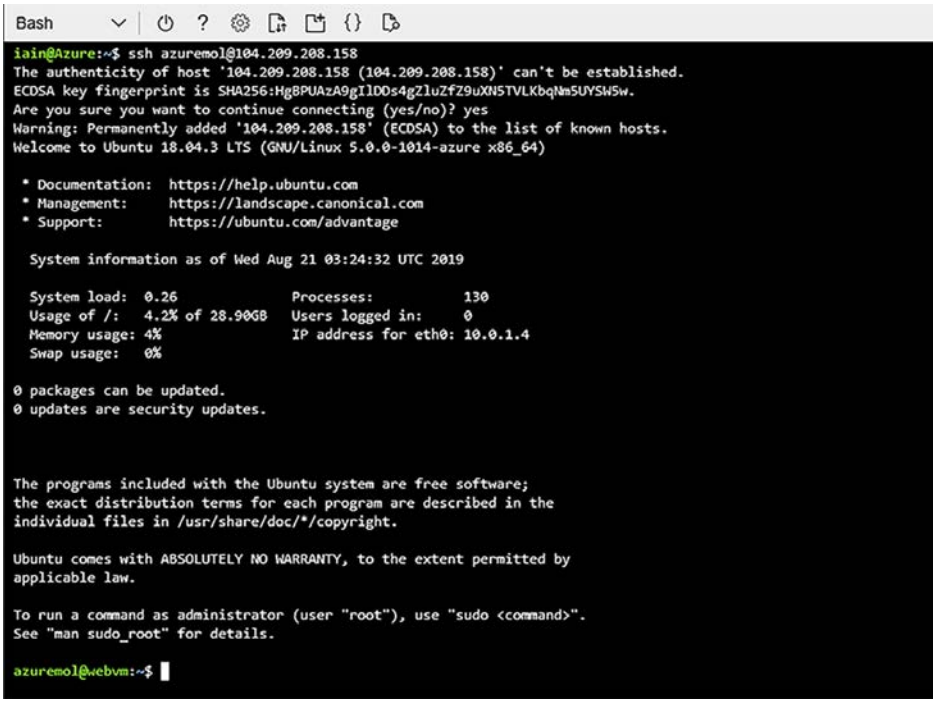


Abbildung 2.5 Verwenden Sie die Verbindungszeichenfolge, die im Azure Portal angezeigt wird, um eine SSH-Verbindung zu Ihrer VM aus der Cloud Shell herzustellen.

Jetzt sind wir am Punkt angelangt, an dem Sie sich entweder wie in Ihrer zweiten „Heimat“ fühlen – oder die Linux-Aufforderung ist völlig fremd für Sie. Macht nichts. Sie müssen keine große Anzahl an Linux-Befehlen kennen, und jeder Befehl wird im Laufe der Aufgabe erklärt. Dennoch empfehle ich Ihnen, zumindest sich einige grundlegende Linux-Verwaltungsfähigkeiten anzueignen. Ein großer Teil der Cloud basiert auf Linux-Systemen, und es gibt eine große Entwicklung in Richtung Container und Mikrodienste bei der Anwendungsentwicklung und -verwaltung. Wenn Sie ein Windows-Admin der alten Schule sind – willkommen! Am Ende des Kapitels wartet diesbezüglich etwas auf Sie, also nur Geduld.

2.4.2 Installation der Webserver

Eine VM erstellen? Erledigt. Verbindung zur VM mit SSH herstellen? Erledigt. Jetzt können Sie die Pakete für einen Webserver installieren und sich darauf vorbereiten, ihn in Aktion zu erleben.

Azure unterstützt viele verschiedene Linux-Distributionen (*Distros*). Paketverwaltungstools und Konfigurationsdateien-Speicherorte variieren ein wenig zwischen den einzelnen Distros. Für dieses Buch wird Ubuntu verwendet, da es eine der beliebtesten und am besten dokumentierten Linux-Distros für das Cloud Computing ist. Wenn Sie also mal festhängen, sollte genügend Dokumentation vorhanden sein, um Ihnen zu helfen, wie https://help.ubuntu.com. Wenn Sie eine andere Distribution verwenden möchten, mit der Sie sich bereits auskennen, können Sie das gerne tun. Bleiben Sie andernfalls bei Ubuntu.

Jetzt testen

Installieren Sie über Ihre SSH-Sitzung mit der VM die Webserverpakete mit APT:

1. In Ubuntu installieren Sie Pakete mit dem Advanced Packing Tool (APT), einem sehr leistungsfähigen Paketverwaltungstool, das automatisch alle zusätzlichen Pakete installiert, die es benötigt. Sie müssen nur „Installiere einen Webserver“ angeben, und APT installiert alle benötigten Komponenten.

   Installieren Sie für dieses Beispiel den LAMP-Webstack. Dies ist wahrscheinlich der häufigste Satz an Webkomponenten: Linux, Apache (ein Webserver), MySQL (ein Datenbankserver) und PHP (eine Web-Programmiersprache):

   ```
   sudo apt update && sudo apt install -y lamp-server^
   ```

   Durch den ersten Befehl werden die verfügbaren Pakete aktualisiert. Dadurch wird sichergestellt, dass Sie die neuesten und tollsten Pakete installieren können. Im Anschluss an diesen Befehl führen Sie den nächsten Befehl mit `&&` aus. Warum nicht einfach eine neue Zeile für den nächsten Befehl beginnen? Durch `&&` wird der nächste Befehl nur ausgeführt, wenn der vorherige Befehl erfolgreich abgeschlossen wurde. Wenn zum Beispiel keine Netzwerkverbindung bestand, damit apt die neuesten Pakete abrufen kann (Bin ich nicht witzig? Ich weiß, dass Netzwerk-Konnektivität vorhanden sein muss, um überhaupt eine Verbindung herstellen zu können!), ergibt es keinen Sinn, den `install`-Befehl auszuführen.

   Wird der update-Befehl erfolgreich ausgeführt, wird durch apt anschließend festgelegt, welche zusätzlichen Pakete benötigt werden und die Installation des `lamp-server` beginnt. Wozu dient das Caret-Symbol am Ende (`^`)? Das Symbol ist die Aufforderung an `apt`, den gesamten Satz von Paketen zu installieren, aus denen sich der AMP-Server zusammensetzt, nicht nur ein einzelnes Paket namens `lamp-server`.

2. Eventuell werden Sie durch das Installationsprogramm zur Eingabe eines Kennworts aufgefordert oder dazu, den Standard eines leeren MySQL-Kennworts zu verwenden. Das ist allerdings nicht sehr sicher, für die tatsächliche Produktion müssen Sie ein sicheres Kennwort angeben. In Kapitel 15 werden Sie richtig cool und speichern ein starkes, sicheres Kennwort in Azure Key Vault, das automatisch in diesen MySQL-Installationsassistenten injiziert wird.

   Es dauert etwa eine Minute, bis alle Pakete für Ihren LAMP-Webstack installiert sind, und dann sind Sie fertig.
3. Geben Sie `exit` ein, um sich bei Ihrer VM abzumelden und zur Cloud Shell-Aufforderung zurückzukehren.

So einfach ist das. Ihr Webserver ist in Betrieb, aber Sie werden noch nicht über einen Webbrowser darauf zugreifen können. Dazu müssen Sie zulassen, dass Webverkehr die VM erreicht.

2.5 Zulassen, dass Webverkehr die VM erreicht

Ihr Webserver ist in Betrieb, aber wenn Sie die öffentliche IP-Adresse Ihrer VM in einem Webbrowser eingeben, wird die Webseite nicht geladen. Warum? Erinnern Sie sich an die Netzwerksicherheitsgruppen, die in Abschnitt 2.1.5 kurz besprochen wurden? Bei Erstellen der VM wurde auch eine Netzwerksicherheitsgruppe für Sie erstellt. Es wurde eine Regel hinzugefügt, durch die eine Remote-Verwaltung ermöglicht wird: in diesem Fall SSH.Der Rest der VM ist gesperrt. Um Besucher*innen den Zugriff auf Ihren Webserver über das Internet zu ermöglichen, müssen Sie eine Regel in der Netzwerksicherheitsgruppe erstellen, die den Webverkehr zulässt. Sonst kann niemand Pizza bestellen!

2.5.1 Erstellen einer Regel zum Zulassen von Webverkehr

In diesem Abschnitt wird es etwas spannender, weil wir die Azure CLI nutzen, um eine Regel für den Webverkehr zu erstellen. Sie hätten diesen HTTP-Port im Portal öffnen können, als Sie die VM erstellt haben, aber dann hätten Sie nur den halben Spaß gehabt!

Die Azure CLI ist in der Cloud Shell erhältlich. Sie müssen nichts installieren. In Kapitel 5 befassen wir uns mit virtuellen Vernetzungen und Netzwerksicherheitsgruppen. Vorerst können wir überprüfen, wie schnell und leistungsstark die Azure CLI mit nur einem Befehl ist.

Jetzt testen

Öffnen Sie die Azure Cloud Shell und führen Sie diese Schritte aus, um die Azure CLI in Aktion zu sehen:

1. Wenn Sie Ihr Cloud Shell-Fenster geschlossen haben, öffnen Sie es wieder über das Azure Portal. Vergewissern Sie sich, dass die Bash-Shell geladen wird, nicht PowerShell. Bei Bedarf wechseln Sie zur Bash-Version.
2. Um die Azure CLI und installierte Module anzuzeigen, geben Sie `az --version` ein. Eine Liste mit Modulen und Versionsnummern wird angezeigt. Das Tolle an der Cloud Shell ist, dass immer die neueste und tollste Version zur Verfügung steht.

HINWEIS Wenn Sie gut aufgepasst haben, ist Ihnen vielleicht aufgefallen, dass der Befehl Informationen über die Version von Python ausgegeben hat.

Warum sind diese Informationen wichtig? Python ist eine leistungsstarke, beliebte Programmiersprache. Die Azure CLI ist in Python geschrieben, wodurch sie plattformübergreifend verwendet werden kann und für Sie zur lokalen Installation auf jedem Computer zur Verfügung steht, wenn Sie die Cloud Shell nicht immer verwenden möchten. Microsoft möchte, dass zur Open Source-Community beigetragen wird. Daher steht die Azure CLI auf GitHub zur Verfügung und alle können Beiträge leisten, Vorschläge machen oder Probleme melden (https://github.com/Azure/azure-cli).

3 Um einen Port zu öffnen, geben Sie den VM-Namen und die entsprechende Ressourcengruppe sowie die Portnummer an. Für den Webverkehr müssen Sie Port 80 öffnen. Geben Sie die Ressourcengruppe (`-g`) und den VM-Namen (`-n`) ein, die Sie bei der Erstellung Ihrer VM angegeben haben:

```
az vm open-port -g azuremolchapter2 -n webvm --port 80
```

2.5.2 Der Webserver in Aktion

Sie haben einen Port für Ihre VM geöffnet und nun sehen wir, was passiert, wenn Sie versuchen, über einen Webbrowser darauf zuzugreifen:

1 Wählen Sie im Azure Portal Ihre VM aus, wenn Sie an eine andere Stelle navigiert haben. Die öffentliche IP-Adresse wird oben rechts in der VM-Übersichtsseite aufgeführt.
2 Wählen Sie die Adresse aus und kopieren Sie sie.
3 Öffnen Sie in Ihrem Webbrowser eine neue Registerkarte oder ein neues Fenster, und fügen Sie die öffentliche IP-Adresse ein. Die Standard-Apache-Website lädt (siehe Abbildung 2.6). Okay, es sieht nicht aus wie bei einem Pizzabäcker, aber Sie haben den Grundstein gelegt, um Ihren Code einzuspeisen und Ihre Anwendung zu entwickeln.

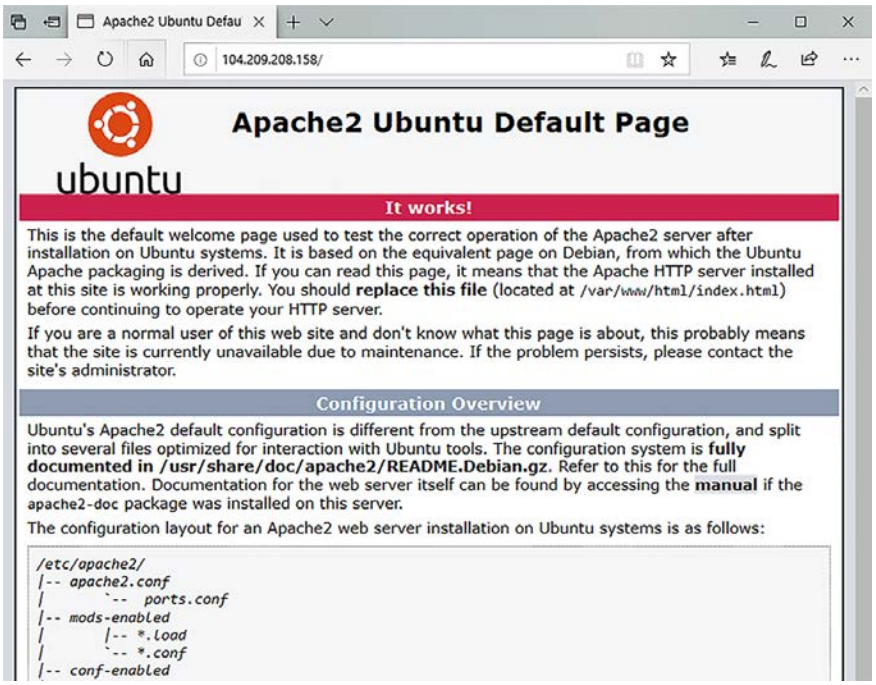


Abbildung 2.6 Um Ihren Webserver in Aktion zu erleben und die Standard-Apache 2-Seite anzuzeigen, geben Sie die öffentliche IP-Adresse in einem Webbrowser ein.

2.6 Übung: Erstellen einer Windows-VM

In den vorherigen Abschnitten sind die Installation des LAMP-Stacks auf einer Ubuntu Linux-VM Schritt für Schritt durchgegangen. Dies ist eine gängige Plattform für Websites, aber vielleicht verspüren Sie den Wunsch nach etwas Liebe und Aufmerksamkeit, wenn Sie einen Windows-Hintergrund haben. Ihre Entwicklungsteams oder geschäftlichen Entscheidungsträger*innen können beispielsweise .NET nutzen. Sie können .NET Core auch auf Linux-VMs ausführen, also lassen Sie Ihre Entscheidung nicht durch die Sprache beeinflussen.

Versuchen Sie anhand des Gelernten aus dem Schritt-für-Schritt-Beispiel, eine VM zu erstellen, die die Internet Information Services (IIS) ausführt. Hier sind einige Tipps:

- Sie benötigen eine VM, die Windows Server 2019 ausführt.
- Sie verwenden RDP, nicht SSH, also wird das Verbindungserlebnis etwas anders ausfallen.
- Suchen Sie im Server Manager nach einer Option, um Rollen und Funktionen hinzuzufügen.
- Sie müssen den Webserver (IIS) installieren.
- Vergessen Sie nicht, einen Netzwerkport für HTTP-Verkehr auf TCP-Port 80 zu öffnen. Sie können das Portal verwenden, wenn Sie möchten.

2.7 Bereinigen von Ressourcen

Während Sie Ressourcen in Azure erstellen, fängt der Abrechnungszähler an zu laufen. Die Abrechnung verläuft minutenweise, daher sollten Sie sich gute Gewohnheiten aneignen und die Ausführung von Ressourcen wie VMs beenden, sobald Sie mit ihnen fertig sind. Sie haben zwei Möglichkeiten, die Abrechnung für das Ausführen einer VM zu stoppen:

- *Deaktivieren einer VM.* Sie können die Stopp-Schaltfläche im Portal auswählen, um eine VM anzuhalten und zu deaktivieren, wodurch alle verwendeten Computing- und Netzwerkressourcen freigegeben werden.
- *Löschen einer VM.* Diese Option erklärt sich selbst. Wenn in Azure nichts mehr vorhanden ist, gibt es auch nichts zu bezahlen. Vergewissern Sie sich, dass Sie mit der VM fertig sind, bevor Sie sie löschen. Es gibt keine Zurück-Schaltfläche in Azure.

Ich empfehle Ihnen, für jede Anwendungsbereitstellung eine Ressourcengruppe zu erstellen, wenn Sie damit beginnen, Dinge in Azure zu erstellen. Beim Durchgehen der Übungen in diesem Buch tun Sie genau das. Wenn Sie Ihre Ressourcengruppen nach Kapiteln benennen, z. B. `azuremolchapter2`, behalten Sie einen besseren Überblick über Ihre Ressourcen und darüber, was zu löschen ist. Diese Übung erleichtert das Bereinigen etwas, denn Sie können bei Abschluss eines Kapitels die gesamte Ressourcengruppe löschen. Wählen Sie also „Resource Groups“ (Ressourcengruppen) im Navigationsmenü auf der linken Seite des Bildschirms aus, öffnen Sie die Ressourcengruppen, die Sie in diesem Kapitel erstellt haben, und wählen Sie dann „Delete Resource Group“ (Ressourcengruppe löschen) – siehe

Abbildung 2.7. Zur Bestätigung werden Sie zur Eingabe des Namens der Ressourcengruppe aufgefordert.

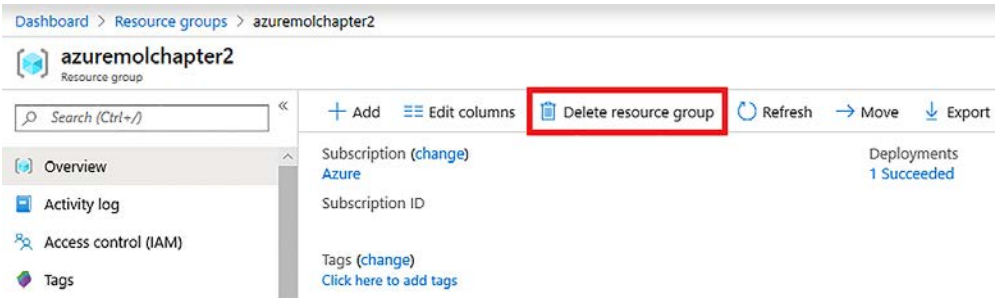


Abbildung 2.7 Um Kosten zu sparen, löschen Sie Ressourcengruppen, wenn Sie sie nicht mehr benötigen.

Wenn Sie sich angewöhnen, Ressourcen zu löschen, sobald Sie mit ihnen fertig sind, können Sie dieses Buch problemlos mit der kostenlosen Azure-Gutschrift durcharbeiten. Deaktivieren Sie Ihre VM am Ende jeder Lektion zumindest, damit Sie am nächsten Tag weitermachen können und dazwischen die Abrechnungszeit anhalten.

2.8 Houston, wir haben ein Problem

Manchmal werden Sie bei Azure auf Probleme stoßen. Ja, ich habe es ausgesprochen. Normalerweise verhält sich die Azure-Plattform gut bei auftretenden Problemen während der Erstellung von Ressourcen:

- Die Azure CLI oder Azure PowerShell gibt Rückmeldung, wenn Sie Befehle ausführen, daher sollte es offensichtlich sein, wenn etwas schief geht. Azure PowerShell verwendet in der Regel einen schönen, ruhigen, roten Text, um Ihre Aufmerksamkeit zu erregen.
- Die Azure CLI kann dabei etwas kryptischer sein, da sie in der Regel die tatsächlichen Reaktionen auf die zugrunde liegenden REST API-Aufrufe vom Server enthält. Wenn das neu ist, können einige erfolgreiche und weniger erfolgreiche Versuche erforderlich sein, um zu verstehen, was schief gelaufen ist. Der hilfreiche Teil der REST-Reaktionen ist, dass Sie die Fehlermeldungen kopieren und in Ihre Lieblingssuchmaschine einfügen können. So erhalten Sie in der Regel zuverlässige Ergebnisse zur Problembehandlung.

Hat Ihnen das den REST gegeben? Wir haben gerade erst angefangen!

Wenn Sie eine Webseite in Ihrem Browser öffnen, kommuniziert Ihr Computer mit einem Webserver über HTTP. Ich bin mir fast sicher, dass Sie schon einmal eine 404-Fehlermeldung auf einer Website gesehen haben. Diese Meldung bedeutet, dass die Webseite nicht gefunden werden konnte. Andere häufige Fehler, die Sie vielleicht schon gesehen haben, sind 403, wenn Sie keine Berechtigung zur Anzeige einer Seite haben, und 500, wenn der Server auf einen Fehler stößt.

(Fortsetzung)

Selbst wenn alles gut läuft, erhält Ihr Browser dennoch Code-200-Nachrichten, wenn die Seite gut lädt, oder Code-301-Nachrichten, wenn eine Seite umgeleitet wurde. Sie müssen alle diese Codes nicht verstehen und nachverfolgen – es handelt sich dabei nur um einen Standardweg, wie HTTP die Kommunikation zwischen Computern erleichtert.

Weiter vorne in diesem Kapitel haben wir schon darüber gesprochen, wie Azure-Ressourcen über das Webportal, die CLI oder PowerShell erstellt und verwaltet werden können. Auf alle Azure-Dienste wird über die Representational State Transfer (REST)-Anwendungsprogrammierschnittstellen (APIs) zugegriffen.

Falls Ihnen das neu ist; REST APIs sind eine (mehr oder weniger …) standardisierte Art, Dienste über HTTP bereitzustellen. Sie verwenden Standard-HTTP-Anfragen wie `GET` und `POST`, um Informationen anzufordern oder Änderungen vorzunehmen, und sobald die Plattform die Anfrage akzeptiert und verarbeitet, erhalten Sie eine Statusmeldung. Azure hat eine genau definierte Reihe von REST APIs.

Sie müssen nicht verstehen, was das alles bedeutet. Seien Sie sich nur bewusst, dass sich Fehlermeldungen nicht immer in einem für Menschen gut lesbaren und hilfreichen Format befinden. Manchmal erhalten Sie die rohe HTTP-Reaktion von der REST API, die Sie selbst entziffern müssen. Und wieder können Sie diesen Fehler in Ihre Lieblingssuchmaschine einfügen. Sie haben eine gute Chance, dass jemand bereits auf das Problem gestoßen ist und eine menschlich verständlichere Erläuterung für das geliefert hat, was schief gelaufen ist und was Sie korrigieren müssen.

Die häufigsten Probleme mit VMs treten auf, wenn Sie sich mit Ihrer VM verbinden. Sie könnten sich für die Remoteadministration mit SSH oder RDP verbinden oder versuchen, über einen Webbrowser oder Desktopclient auf Ihre Anwendungen zuzugreifen. Diese Probleme sind oft netzwerkbezogen. Ich hebe es mir für Kapitel 5 auf, den Netzwerkern die Schuld in die Schuhe zu schieben, hier also einige Prüfpunkte:

- Können Sie eine Verbindung zu anderen Azure-VMs oder Anwendungen herstellen, die auf Azure ausgeführt werden? Wenn nicht, verhindert vermutlich ein lokaler Aspekt in Ihrem Netzwerk den Zugriff.

  Wenn Sie eine Verbindung zu anderen Azure-Ressourcen herstellen können, stellen Sie sicher, dass Sie die Regeln der Netzwerksicherheitsgruppe geöffnet haben (Abschnitt 2.5). In Kapitel 5 werden diese Regeln näher erläutert.
- Bei Authentifizierungsproblemen versuchen Sie Folgendes:
  - Vergewissern Sie sich, dass Sie über die richtigen SSH-Schlüssel verfügen. Azure sollte Ihnen beim Erstellen der VM mitteilen, ob der öffentliche Schlüssel ungültig ist, doch wenn Sie über mehr als einen privaten Schlüssel verfügen, stellen Sie sicher, dass Sie den richtigen verwenden.
  - Versuchen Sie bei RDP-Problemen eine Verbindung zu localhost\<*username*> herzustellen, und geben Sie Ihr Kennwort ein. Standardmäßig versuchen die meisten RDP-Clients, lokale oder Netzwerkanmeldedaten zu präsentieren, die Ihre VM nicht versteht.

3 Azure Web Apps

In Kapitel 2 haben Sie eine virtuelle Maschine (VM) erstellt und manuell Pakete installiert, um einen grundlegenden Webserver ausführen zu können. Mit dieser VM könnten Sie die Website für den Pizzabäcker erstellen, wenn Sie schon Hunger haben und loslegen möchten. Einer der größten Anwendungsfälle für Azure-VMs ist die Ausführung von Webanwendungen, in der Regel im großen Umfang. Webanwendungen sind gut geeignete Workloads für VMs. Schön und gut, wenn Sie auch den Wartungsaufwand mögen, der mit der Verwaltung all dieser VMs einhergeht. Sie wissen schon ... unterhaltsame Dinge wie Softwareupdates, Sicherheitspatches, zentralisierte Protokollierung und Compliance-Berichte. Was wäre, wenn Sie zur Ausführung Ihrer Webanwendungen das gesamte Potenzial eines sicheren Webservers nutzen könnten, einschließlich der automatischen, bedarfsgerechten Skalierung, doch ohne die Notwendigkeit, all diese VMs zu erstellen und zu verwalten? Hier stelle ich Ihnen den Azure Web Apps-Dienst vor.

In diesem Kapitel vergleichen wir den „Infrastructure as a Service"-Ansatz (IaaS) von VMs und Webservern mit dem „Platform as a Service"-Ansatz (PaaS). Sie lernen die Vorteile von Azure Web Apps beim Erstellen einer Webanwendung kennen und sehen, wie Sie mit den Entwicklungs- und Produktionsveröffentlichungen arbeiten. Anschließend lernen Sie, wie Sie Ihre Web App automatisch über eine Quellcodeverwaltung wie GitHub bereitstellen können. Dieser Workflow wird in Abbildung 3.1 gezeigt. Mit Azure Web Apps können Sie Ihre Pizzabäcker-Website

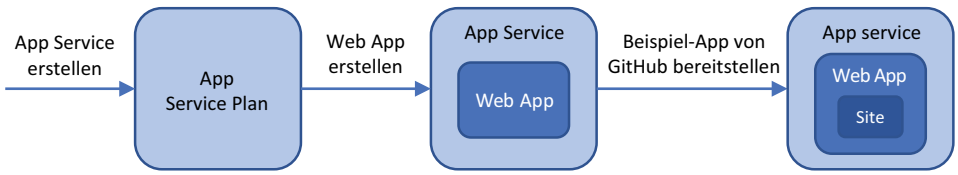


Abbildung 3.1 In diesem Kapitel erstellen Sie einen App Service Plan sowie eine grundlegende Web App und stellen dann über GitHub eine Website bereit.

innerhalb weniger Minuten bereitstellen und ausführen, ohne eine VM und Webserverpakete installieren und konfigurieren zu müssen.

3.1 Azure Web Apps – Übersicht und Konzepte

Mit Azure Web Apps erhalten Sie einen Einblick in die wunderbare Welt der PaaS-Lösungen. Wenn Sie denken, bei Cloud Computing geht es nur um VMs, sollten Sie diese Idee wahrscheinlich ein wenig überdenken. Zu Beginn dieses Buchs sprach ich über das Erwerben von Rechenleistung und die Fokussierung auf Ihre Anwendungen und Kunden. Beim Wechsel von IaaS-Lösungen, wie VMs, zu PaaS-Lösungen, wie Web Apps, werden Ihre Anwendungen und Kund*innen in den Mittelpunkt gerückt.

Für die Ausführung von Webanwendungen erfordern VMs die Verwaltung des Betriebssystems, von Anwendungsupdates und Sicherheits- sowie Verkehrsregeln und die Konfiguration des gesamten Systems. Mit Web Apps laden Sie Ihre Webanwendung hoch, und alle diese Verwaltungsaufgaben werden für Sie erledigt. Jetzt können Sie sich darauf konzentrieren, die Anwendungserfahrung für Ihre Kund*innen oder die Verfügbarkeit durch Skalierungsmöglichkeiten und Verkehrsverwaltung zu verbessern.

Sollten Sie also VMs niemals für das Hosting einer Webanwendung ausführen? Wahrscheinlich nicht. Es gibt gute Gründe dafür, den gesamten Anwendungsstack auszuführen und selbst zu konfigurieren, z. B. wenn Sie sehr spezifische Anwendungsunterstützung oder Sprachlaufzeit benötigen. Doch Web Apps deckt viele der Anwendungsfälle für das Ausführen von Webanwendungen ab.

3.1.1 Unterstützte Sprachen und Umgebungen

Welche Flexibilität bietet Web Apps in Bezug auf Programmiersprachen, mit denen Sie Ihre Webanwendung entwickeln können? Ziemlich viel! Es gibt zwei primäre Plattformen zum Ausführen von Web Apps: Windows und Linux. Sie können .NET Core, Node.js, Python, Java, Ruby und PHP-Webanwendungen sowohl auf Windows- als auch auf Linux-Webanwendungsinstanzen nativ ausführen. Unter Windows können Sie auch das gesamte .NET Framework ausführen. Wenn Sie wirklich cool sein und Ihre Webanwendung in Containern ausführen möchten, gibt es Web Apps auch für Container, mit denen Sie native Docker-Container für Linux ausführen können. In Kapitel 19 gehen wir mehr auf Container und Docker ein. Zunächst sollten Sie verstehen, dass Ihre Optionen mit Web Apps abgedeckt sind.

Wann macht die Nutzung von Web Apps eventuell keinen Sinn? Nicht alle Anwendungssprachen werden von Web Apps unterstützt. Angenommen, Sie möchten sich wirklich mit einer Web-Applikation in Perl herumschlagen. In dem Fall würden Sie wahrscheinlich auf IaaS-VMs zurückgreifen, die Sie selbst verwalten, da Web Apps Perl nicht unterstützt. Web Apps unterstützt aber die wohl gängigsten Web-Programmiersprachen, die Sie verwenden möchten. Sie sollten sich wahrscheinlich auch eine neuere Version Ihrer App ansehen, als eine, die in Perl geschrieben ist.

Web Apps unterstützt nicht nur verschiedene Sprachen, sondern auch verschiedene Versionen dieser Sprachen. Zum Beispiel PHP: Es gibt in der Regel drei oder vier Versionen von PHP, aus denen Sie wählen können, um Ihre Anwendung ideal zu unterstützen. Und das Beste daran ist, dass Sie sich im Gegensatz zur Verwaltung einer IaaS-Plattform keine Gedanken um die Abhängigkeiten auf dem zugrunde liegenden Webserver machen müssen, um alles zu unterstützen. Python ist ein weiteres Beispiel für Unterschiede zwischen den stabilen Versionen 2.7 und 3.6 und höher – siehe Abbildung 3.2.

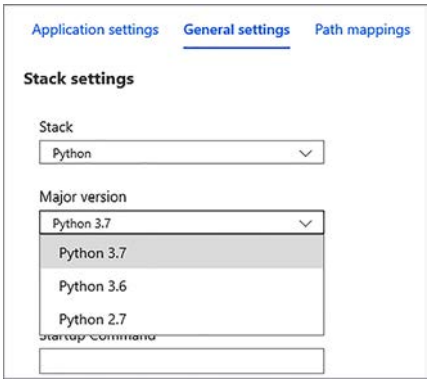


Abbildung 3.2 Wählen Sie eine bestimmte Version einer Sprache in den Einstellungen der Web Apps-Anwendungen aus.

Mit Web Apps bleiben Sie auch bei Sicherheitskorrekturen auf dem neuesten Stand. Sie sollten aber nicht davon ausgehen, dass eine ältere Version von PHP oder Python weiterhin unterstützt wird. Ab einem bestimmten Punkt wird es eine Kürzung der unterstützten älteren Versionen geben. Wieder kann es sich dabei um einen Punkt handeln, an dem Sie auf die eigenständige Ausführung von IaaS-VMs zurückgreifen, wenn Ihre App eine ältere Sprachversion benötigt. Wenn Sie allerdings eine ältere Version einer bestimmten Sprache ausführen müssen, um eine ältere Anwendung zu unterstützen. Lassen Sie sich nicht auf einen konstanten Wartungsansatz ein. Achten Sie immer darauf, mit diesen älteren Apps auf modernere unterstützte Plattformen zu wechseln.

3.1.2 Staging unterschiedlicher Versionen mit Bereitstellungsslots

Bereitstellungsslots bieten eine Staging-Umgebung für Ihre Webanwendung. Sie können neue Versionen Ihrer App per Push auf einen Bereitstellungsslot verschieben und sie mit Umgebungsvariablen oder Datenbankverbindungen ausführen, ohne die Livewebsite zu beeinträchtigen. Wenn Sie mit der Anzeige und Verwendung des Bereitstellungsslots zufrieden sind, können Sie diese Version im Handumdrehen auf der Livewebsite bereitstellen. Die vorherige Livewebsite wechselt dann zu einem eigenen Bereitstellungsslot und stellt eine archivierte Version dar. Sie können die App bei Bedarf auch wieder in die Produktion aufnehmen.

Die Anzahl der verfügbaren Bereitstellungsslots variiert je nach der von Ihnen gewählten Ebene der Web App. Durch eine größere Anzahl von Bereitstellungsslots können mehrere Staging-Versionen von verschiedenen Entwickler*innen verwendet werden, während die eigenen Updates inszeniert und getestet werden.

3.1.3 App Service-Pläne

Web Apps ist Teil der breiteren App Service-Familie von Azure. Azure App Service (inklusive Mobile Apps, API Apps, Logic Apps). Alle diese Dienste außer Logic Apps sind in jeder Region verfügbar, in der Azure ausgeführt wird. Eine gute Ressource, um die Verfügbarkeit von Azure-Diensten nach Regionen zu überprüfen finden Sie unter https://azure.microsoft.com/regions/services. Viele Dienste sind global verfügbar.

Wenn Sie eine App Service-Ressource wie eine Web App erstellen oder verwenden müssen, erstellen Sie einen Service-Plan oder verwenden Sie einen bestehenden. Durch den Service-Plan wird die Menge der Ressourcen definiert, die Ihnen zur

Verfügung stehen, sowie wie viel Automatisierung möglich ist, um Ihre Web App zu skalieren und zu sichern, und wie hochverfügbar Ihre Website mit Staging-Slots und Traffic Manager sein sollte (eine Möglichkeit, den Datenverkehr geografisch an die nächstgelegene Instanz für eine*n Benutzer*in weiterzuleiten, die wir in Kapitel 11 betrachten). Wie immer erhalten Sie das, wofür Sie bezahlen. Ihre Anwendungs- und Geschäftsanforderungen sollten Ihnen als Richtschnur dienen, wenn es darum geht, wie viele Ressourcen benötigt werden und welche zusätzlichen Funktionen erforderlich sind. Jede Serviceebene baut auf den Funktionen der darunterliegenden Ebenen auf und es kommen in der Regel mehr Storage und verfügbare Ressourcen hinzu.

Die vier wichtigsten Service Plan-Stufen sind:

- *Free/Shared* – gemeinsame Infrastruktur, minimaler Storage und keine Optionen für die Bereitstellung verschiedener Staging-Versionen, Routing-Optionen für Traffic oder Sicherungen. Mit der Shared-Ebene können Sie eine eigene Domäne verwenden, die Ihnen in Rechnung gestellt wird.
- *Basic* – spezielle Computing-Ressourcen für Ihre Web App. Sie können SSL verwenden und die Anzahl der von Ihnen ausgeführten Web App-Instanzen manuell skalieren. Die Free/Shared- und Basic-Stufen bieten eine gute Umgebung, um den Web Apps Service zu testen. Ich würde Ihnen aber nicht empfehlen, tatsächliche Produktions- oder Entwicklungs-Workloads auszuführen. Die Leistung ist kein einschränkender Faktor, aber Sie fehlen einige der automatisierten Funktionen, wie Sicherungen und Skalierung.
- *Standard* – tägliche Backups, automatische Skalierung von Web App-Instanzen und Bereitstellungsslots und Weiterleitung von Benutzenden mit Traffic Manager. Diese Stufe eignet sich für Anwendungen mit geringem Bedarf oder für Entwicklungsumgebungen, in denen Sie keine große Anzahl von Sicherungen oder Bereitstellungsslots benötigen.
- *Premium* – häufigere Sicherungen, mehr Storage und größere Anzahl an Bereitstellungsslots und Möglichkeiten zur Instanzskalierung. Diese Stufe eignet sich für die meisten Produktions-Workloads.

Der Fall für die Isolation

Mit PaaS-Lösungen wie Web Apps wird die Infrastruktur absichtlich abstrahiert. Wie die Namen einiger der Service Plan-Stufen schon vermuten lassen, werden Web Apps über eine gemeinsame Plattform verfügbarer Ressourcen ausgeführt. Das heißt überhaupt nicht, dass Web Apps nicht sicher sind und andere Ihre privaten Daten einsehen können. Doch aus Compliance- oder regulatorischen Gründen müssen Sie Ihre Anwendungen unter Umständen in einer kontrollierten, isolierten Umgebung ausführen. Hier kommen *App Service-Umgebungen* ins Spiel: Isolierte Umgebungen, in denen Sie App Service-Instanzen wie Web Apps in einem segmentierten Teil eines Azure-Rechenzentrums ausführen können. Sie steuern den ein- und ausgehenden Netzwerkdatenverkehr, können Firewalls implementieren und auf Ihren Ressourcen vor Ort virtuelle private Netzwerkverbindungen (VPN) einrichten.

Alle diese Infrastrukturkomponenten sind noch weitgehend von App Service-Umgebungen abstrahiert, doch dieser Ansatz bietet ein gutes Gleichgewicht, wenn Sie die Flexibilität von PaaS-Lösungen nutzen, aber auch einige der feineren Steuerungselemente bezüglich des Datenverkehrsflusses der Netzwerkverbindungen beibehalten möchten.

Mit den Stufen Free und Basic können Sie schon einiges erreichen, doch für die Produktion sollten Sie die Standard- oder Premium-Stufe verwenden. Im Beispiel dieses Kapitels wird die Standard-Ebene verwendet, damit Ihnen alle verfügbaren Funktionen angezeigt werden. Wenn Sie Web Apps mit Ihren eigenen Anwendungen verwenden, können Sie entscheiden, wie viele dieser Funktionen Sie benötigen, und Sie wählen dementsprechend die am besten geeignete Service Plan-Stufe.

3.2 Erstellen einer Web App

Mit ein wenig Theorie im Hinterkopf sehen wir uns nun eine Web App in Aktion an. Um eine Anwendung auszuführen, sind mehrere Schritte erforderlich. Zuerst erstellen Sie die grundlegende Web App, und Ihnen wird die Standardwebsite in Ihrem Browser angezeigt. Dann verwenden Sie eine Beispielwebseite von GitHub und verschieben diese per Push in Azure. Vielleicht haben Ihre Webentwickler*innen bereits damit begonnen, ein Frontend für Ihre Pizzabäcker-Website zu entwickeln, sodass eine grundlegende Website zum Upload bereitsteht.

> **HINWEIS** Wenn Sie Git noch nie verwendet haben – keine Sorge. Sie müssen an diesem Punkt nicht verstehen, was Git macht, am Ende des Kapitels ist Raum zum Experimentieren und Entdecken. *Nutzen Sie Ihre Mittagspause, um mehr über Git zu erfahren,* von Rick Umali (https://www.manning.com/books/learn-git-in-a-month-of-lunches), ist ein hervorragender Einstieg in die Verwendung von Git, wenn Sie etwas mehr darüber erfahren möchten. Es kann kostenlos auf der Manning liveBook-Plattform gelesen werden.

3.2.1 Erstellen einer grundlegenden Web App

Genau wie in Kapitel 2 werde ich Ihnen einige allgemeine Anleitungen mit auf den Weg geben. Finden Sie jedoch heraus, ob Sie einige der Theorien über Anwendungslaufzeiten und App Service-Pläne anwenden können, um eine Web App zu erstellen. Wenn Sie sich bei einigen Optionen nicht sicher sind, können Sie vorerst die Standardeinstellungen übernehmen.

PaaS, nicht IaaS

Diese Web App ist eine neue Ressource und unabhängig von VMs wie denen, die Sie in Kapitel 2 erstellt haben. Dies ist ein IaaS-Ansatz für die Entwicklung und den Betrieb von Webanwendungen. Der PaaS-Ansatz bezieht sich auf Web Apps. Es gibt keinen direkten Bezug zwischen den beiden Typen. Wenn Sie den Ratschlägen in Kapitel 2 gefolgt sind und Ihre VM gelöscht haben, wird diese Web App sogar ganz ohne VM in Ihrem Azure-Abonnement ausgeführt.

Jetzt testen

Um Ihre Web App zu erstellen, führen Sie die folgenden Schritte aus:

1. Öffnen Sie https://portal.azure.com in einem Webbrowser, und melden Sie sich bei Ihrem Azure-Konto an.
2. Sobald Sie sich angemeldet haben, wählen Sie im Portal in der oberen linken Ecke des Dashboards „Create a Resource“ (Eine Ressource erstellen).
3. Wählen Sie „Web“ aus der Liste der Ressourcen aus, die Sie erstellen können, und wählen Sie dann „Web App“.
4. Damit alles übersichtlich und gut organisiert wie in Kapitel 2 ist, schlage ich vor, dass Sie eine dedizierte Ressourcengruppe für Ihre Web App erstellen, „azuremolchapter3“.

5. Geben Sie für den Namen der Web App einen global einmaligen Namen ein. Dieser Name muss global einzigartig sein, da über ihn die URL zu Ihrer Web App im Format http://<*name*>.azurewebsite.net erstellt wird. Falls Sie sich das gefragt haben: Ja, Sie können hier einen eigenen Domänennamen verwenden. Verwenden Sie vorerst die Standardadresse; azurewebsites.net.
6. Sie werden einen einfachen HTML-Code pushen, keinen Docker Container, aber schauen Sie sich all die verschiedenen Runtime-Stacks an, die zur Verfügung stehen. Sie können diese Einstellung ändern, nachdem Sie die Web App erstellt haben. Wählen Sie jedoch zunächst eine ASP.NET-Laufzeit, die unter Windows ausgeführt wird.
7. Lassen Sie Azure automatisch einen App Service Plan erstellen, aber ändern Sie die Größe auf S1 Standard. Diese Stufe bietet alle Kernfunktionen, ohne dass besonders viele Ressourcen für Ihre grundlegende Demowebsite zur Verfügung gestellt werden. In der Praxis können Sie in diesem Schritt Ihre eigenen App Service-Pläne manuell erstellen und konfigurieren oder einen vorhandenen Plan auswählen.
8. Wenn Sie bereit sind, überprüfen und erstellen Sie Ihre erste Web App.

Es nimmt einige Sekunden in Anspruch, Ihren App-Dienst zu erstellen. Wenn Sie bereit sind, wählen Sie in der Navigationsleiste auf der linken Seite des Bildschirms „App Services“ (App-Dienste) aus, und wählen dann Ihre Web App aus der Liste. Über das Übersichtsfenster Ihrer Web App können Sie die URL der Web App anzeigen und auswählen, wie https://azuremol.azurewebsites.net.

Wenn Sie die URL zu Ihrer Web App auswählen, wird ein neues Browser-Fenster oder eine neue Registerkarte geöffnet. Die Standard Web App-Seite lädt (siehe Abbildung 3.3). Sieht immer noch nicht nach Pizza aus. . . .

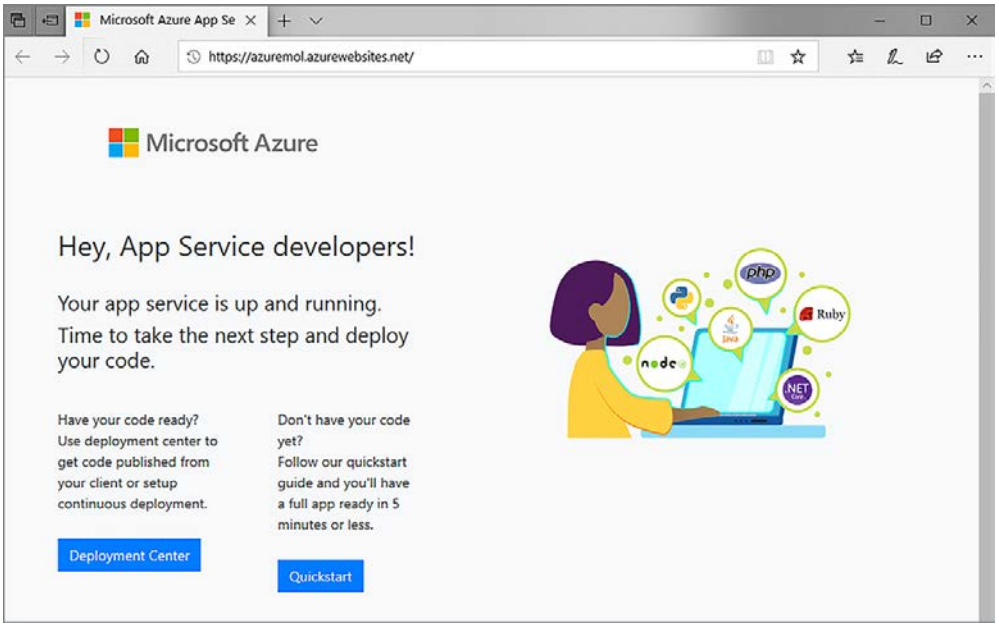


Abbildung 3.3 Um die Standard Web App-Seite in Aktion zu erleben, öffnen Sie die URL Ihrer Website in einem Webbrowser.

3.2.2 Bereitstellen einer HTML-Beispielseite

Sie haben eine Web App auf Azure, aber es ist die langweilige Standardwebseite. Wie erhalten Sie Ihre eigene Website in Azure? Eine der häufigsten plattformübergreifenden Möglichkeiten ist es, Git zu verwenden. Die meisten Anwendungsentwickler*innen und -teams verwenden ein Quellcodeverwaltungssystem. Anstatt Dateien auf Ihrem Computer aufzubewahren und Änderungen im Verlauf der Bearbeitung zu speichern, verfolgen Quellcodeverwaltungssysteme Änderungen nach, und Sie können gemeinsam mit anderen arbeiten. Sie können Testveröffentlichungen erstellen, die Ihren Produktionscode nicht beeinflussen, und auf frühere Versionen zurückgreifen, falls Probleme auftreten. Git ist eines der am häufigsten verwendeten Quellcodeverwaltungssysteme und GitHub ist ein cloudbasierter Dienst, mit dem Sie Code mit dem Rest der Welt teilen und dazu beitragen können. Microsoft hat GitHub 2018 übernommen, aber es gibt keine Verpflichtungen für Sie zwischen GitHub und Azure. Alle Beispiele in diesem Buch sind über GitHub erhältlich.

Für dieses Beispiel erstellen Sie eine lokale Kopie der statischen HTML-Beispielseite und verschieben die Dateien dann per Push auf Ihre Azure Web App. Dieser Workflow wird in Abbildung 3.4 gezeigt.

Abbildung 3.4 Sie erstellen eine lokale Kopie der Beispieldateien von GitHub mit dem Befehl `git clone`. Um diese lokalen Dateien per Push in Ihre Azure Web App zu verschieben, verwenden Sie `git push`.

Jetzt testen

Um eine Kopie der HTML-Beispielseite von GitHub zu erhalten und per Push in Ihre Web App zu verschieben, führen Sie die folgenden Schritte aus:

1. Öffnen Sie die Cloud Shell im Azure Portal, und warten Sie einige Sekunden, bis eine Verbindung für Ihre Session hergestellt wird. Für den Einstieg benötigen Sie die HTML-Beispielseite von GitHub.

   Zum *Klonen* oder Kopieren der HTML-Beispielseite von GitHub geben Sie folgenden Befehl ein:

   ```
   git clone https://github.com/foulDsy/azure-mol-samples-2nd-ed.git
   ```

   Wenn dies Ihre erste Verwendung mit Git in der Cloud Shell ist, müssen Sie einige Einstellungen vornehmen, damit Git versteht, wer Sie sind. Bei den meisten Übungen in diesem Buch spielt es keine Rolle, aber für den Einsatz mit eigenen Projekten und Anwendungen ist es eine gute Möglichkeit, um zu verfolgen, wer bestimmte Aktionen in einem Quellcodeverwaltungssystem durchführt. Sie müssen diese Einstellungen nur einmal vornehmen. Geben Sie Ihre eigene E-Mail-Adresse und den vollständigen Namen wie folgt unter git `config` ein:

```
git config --global user.email "iain@azuremol.com"
git config --global user.name "Iain Foulds"
```

2 Wechseln Sie in das Verzeichnis „azure-mol-samples-2nd-ed“, das erstellt wurde, als Sie das Git-Repository geklont haben:

```
cd azure-mol-samples-2nd-ed/03/prod
```

3 Um sich auf den Upload der HTML-Beispielseite vorzubereiten, müssen Sie Git initialisieren und dann Ihre Dateien hinzufügen und übertragen. Machen Sie sich jetzt keine allzu großen Gedanken um die Git-Befehle. Sie müssen Git mitteilen, welche Dateien verfolgt und hinzufügt werden sollen, und eine Möglichkeit für sich selbst finden, diese Änderungen später nachzuverfolgen, wenn nötig:

```
git init && git add . && git commit -m "Pizza"
```

4 Jetzt können Sie diese HTML-Beispielseite per Push in Ihre Web App verschieben. Definieren Sie zuerst die Anmeldedaten für die Bereitstellung. Um Web Apps zu sichern, wenn Sie eine Bereitstellungsmethode wie Git oder FTP verwenden, müssen Sie einen Benutzungsnamen und ein Kennwort angeben. Web Apps können eine Reihe von Anmeldedaten verwenden, die für alle App Service-Pläne in Azure gültig sind, oder Anmeldedaten auf App-Ebene, die nur für eine einzelne App gelten.

   In der Praxis empfehle ich die Verwendung von Anmeldedaten auf Anwendungsebene, um die Reichweite eines Angriffs zu minimieren, falls Anmeldedaten offengelegt werden. Azure generiert die Anmeldedaten auf App-Ebene automatisch, aber Sie müssen diese jedes Mal abrufen und zuweisen. Um die Sache einfach zu halten, verwenden Sie definierte Anmeldedaten, den Sie in den nächsten Kapiteln wiederverwenden können.

   Erstellen Sie die Anmeldedaten für die Bereitstellung, und geben Sie Ihren eigenen Benutzungsnamen und ein sicheres Kennwort an. Der Benutzungsname muss global eindeutig sein. Wenn es hilft, fügen Sie Ihre Initialen zum Benutzungsnamen hinzu oder eine Namenskonvention, die für Ihre Umgebung sinnvoll ist.

```
az webapp deployment user set --user-name azuremol --password @azurem0l!
```

5 Als Nächstes müssen Sie die URL für das Git-Repository Ihrer Web App abrufen. Geben Sie den Namen der Web App (nicht den Benutzungsnamen, den Sie in Schritt 4 erstellt haben) und die Ressourcengruppe ein, die Sie bei der Erstellung der Web App angegeben haben, um die URL des Git-Repository anzuzeigen.

Slashdot für Sie selbst

Im folgenden Beispiel und späteren Kapiteln bedeutet der Backslash (\), dass der Befehl auf der nächsten Zeile weitergeht. Es ist eine Möglichkeit, lange Zeilen zu verbinden, und dieser Ansatz wird in einer Menge von Online-Beispielen verwendet, wo Sie die Befehle kopieren und einfügen können. Sie müssen die Backslashes in den Beispielen dieses Buchs nicht eingeben, wenn Sie das nicht möchten. Tippen Sie einfach die zusätzlichen Parameter als Teil einer langen Zeile ein.

Wenn Sie die Windows-Befehlsaufforderung statt einer Bash-Shell verwenden, geben Sie die Backslashes nicht ein. Andernfalls erzielen Sie sicher nicht die gewünschten Ergebnisse.

```
az webapp deployment source config-local-git \
--name azuremol \
--resource-group azuremolchapter3 -o tsv
```

6 Ihre Web App ist so konfiguriert, dass sie mit Git-Repositorys funktioniert, also müssen Sie der Cloud Shell mitteilen, was das entsprechende Repository ist. In Git definieren Sie diese Speicherorte als *remotes*.

Kopieren Sie Ihre Git-Klon-URL aus Schritt 5, und stellen Sie diese URL mit folgendem Befehl als Ziel für die HTML-Beispielseite in der Cloud Shell ein:

```
git remote add azure your-git-clone-url
```

7 Um Dateien mit Git hochzuladen oder zu kopieren, verwenden Sie *Push*. Wohin leitet Git diese per Push weiter? An ein *remote,* wie Sie es in Schritt 6 konfiguriert haben, z. B. *azure*. Der letzte Teil des Befehls ist eine Verzweigung, typischerweise *master*. Mit einer Verzweigung in Git behalten Sie den Überblick über verschiedene laufende Modelle. Eine bewährte Vorgehensweise in Produktionsumgebungen ist es, die Freigabe von Zweigen per Push durchzuführen, die Sie nach Belieben benennen können, z. B. *dev* oder *staging*. Diese zusätzlichen Verzweigungen ermöglichen es, dass Ihr Produktionscode normal ausgeführt wird. Sie können dann sicher und ohne Auswirkungen auf Workloads, die Ihre Kund*innen tatsächlich nutzen, an neuen Funktionen oder Aktualisierungen arbeiten.

Leiten Sie die HTML-Beispielseite per Push zu Ihrer Web App:

```
git push azure master
```

8 Geben Sie bei Aufforderung das Kennwort ein, das Sie für die Anmeldedaten für die Bereitstellung erstellt haben. Sie können das Kennwort kopieren und einfügen, um hier Fehler zu vermeiden.

Sie können an der Ausgabe sehen, dass die bestehende Seite der Standard Web App-Seite entfernt wurde und die HTML-Beispielseite hochgeladen und so konfiguriert wird, dass sie ausgeführt wird. Hier einige Beispielausgaben:

```
Counting objects: 3, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 510 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
remote: Updating branch 'master'. remote: Updating submodules.
remote: Preparing deployment for commit id 'dda01e9d86'.
remote: Generating deployment script.
remote: Generating deployment script for Web Site
remote: Running deployment command...
remote: Handling Basic Web Site deployment.
remote: Creating app_offline.htm
remote: KuduSync.NET from: 'D:\home\site\repository' to:
     'D:\home\site\wwwroot'
remote: Deleting file: 'hostingstart.html'
remote: Copying file: 'index.html'
remote: Deleting app_offline.htm
remote: Finished successfully.
remote: Running post deployment command(s)...
remote: Deployment successful.
To https://azuremolikf@azuremol.scm.azurewebsites.net/azuremol.git
  * [new branch]       master -> master
```

Um Ihre aktualisierte Web App zu sehen, aktualisieren Sie Ihre Website in einem Webbrowser, oder öffnen Sie sie wieder über das Übersichtsfenster im Azure Portal. Es sollte wie das wunderbare Beispiel in Abbildung 3.5 aussehen. Ja, die Site ist einfach, aber der Workflow für die Bereitstellung der grundlegendsten statischen HTML-Website in einer komplexen .NET oder Node.js Web App ist der gleiche.

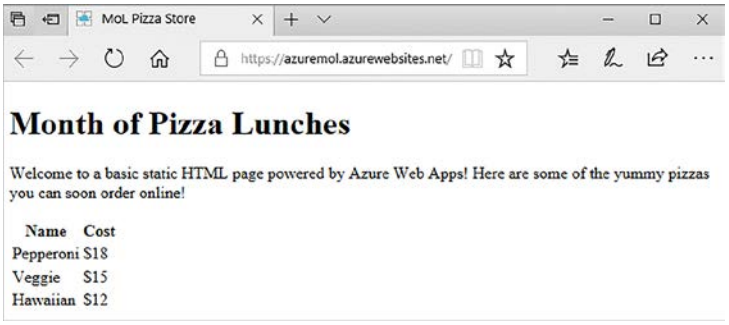


Abbildung 3.5 Aktualisieren Sie Ihren Webbrowser und Sie werden sehen, dass die Standard Web App-Seite durch die grundlegende statische HTML-Site aus GitHub ersetzt wurde.

3.3 Anzeigen von Diagnose-Log-Dateien

Jetzt wissen Sie, wie eine grundlegende Web App erstellt und eine einfache HTML-Site darin bereitgestellt wird, doch was ist mit der allgemeinen Verwaltung? Wenn Sie Probleme haben, wäre es hilfreich, die Webserver- oder Anwendungs-Log-Dateien anzusehen. Zur Unterstützung bei der Fehlerbehebung für Ihre Apps können Sie die Ausgabe Ihrer App in diese Log-Dateien schreiben. Log-Dateien können in Echtzeit angesehen oder in Log-Dateien geschrieben und später überprüft werden.

Ihre Web App wird weitgehend selbstständig ausgeführt. Aus Wartungssicht können Sie auf dem zugrunde liegenden Webhost nicht viel tun. Wenn bei Ihrer Anwendung Probleme auftreten, können Sie sich die Log-Dateien ansehen, um herauszufinden, was los ist, und das Problem zu beheben. Mit Azure Web Apps konfigurieren Sie z. B. die Ebene der Log-Dateinachrichten zur Überprüfung, wo die Log-Dateien gespeichert und wie lange Sie aufbewahrt werden. Abbildung 3.6 skizziert, wie Sie Log-Dateien mit Web Apps generieren und anzeigen.

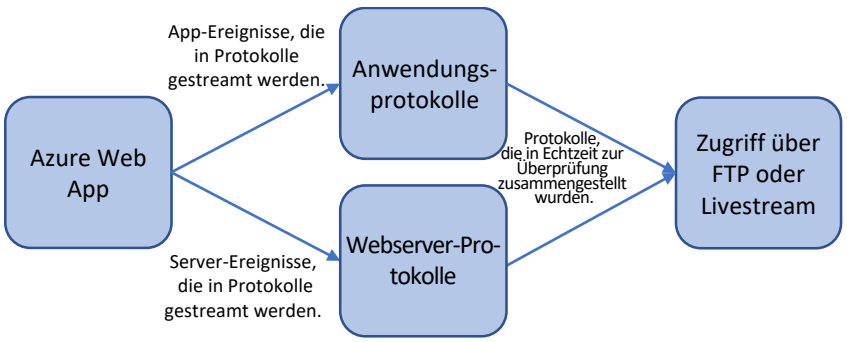


Abbildung 3.6 Ihre Anwendung kann Anwendungs- und Server-Log-Dateien erstellen. Zum Prüfen und Beheben von Problemen können Sie diese Log-Dateien per FTP herunterladen oder in Echtzeit ansehen.

Jetzt testen

Um Ihre Web App für Diagnose-Log-Dateien zu konfigurieren, führen Sie die folgenden Schritte aus.

1. Wählen Sie im Azure Portal die Web App, die Sie in der vorherigen Übung erstellt haben.
2. Im Übersichtsfenster scrollen Sie in den Bereich „Monitoring“ (Überwachung) und wählen Sie „App Service Logs“ (App-Service-Log-Dateien).
3. Überprüfen Sie die verfügbaren Log-Dateioptionen, beispielsweise die Ausführlichkeit, und ob Sie die Verfolgung fehlgeschlagener Anfragen aktivieren möchten. Wenn Sie sich mit der Infrastrukturseite von Azure befassen, müssen Sie möglicherweise mit Ihren Anwendungsentwickler*innen zusammenarbeiten, um festzustellen, welche Log-Dateien sie benötigen, um Probleme zu beheben. Dann können Sie die entsprechende Protokollführung einschalten. Die Protokolle können im lokalen Dateisystem der Web App gespeichert oder zur Verarbeitung mit einer anderen Anwendung in Azure Storage übertragen werden.
4. Aktivieren Sie jetzt die Anwendungsprotokollierung (Dateisystem). Aktivieren Sie auch die Webserverprotokollierung im Dateisystem mit einer Aufbewahrungsfrist von sieben Tagen. Die Standardfehlerebene zeigt möglicherweise nichts an, wenn alles richtig funktioniert. Seien Sie aber vorsichtig, wenn Sie zu Debug oder Trace wechseln, da sich Ihre Protokolle schnell füllen können und es schwierig ist, tatsächlich zu sehen, was passiert! Wenn Sie ein Problem haben, erhöhen Sie schrittweise die Protokollierebene, bis Sie genügend Informationen zur Fehlerbehebung erfasst haben, ohne von den Protokolldaten überfordert zu sein.

Wenn Sie sich wirklich mit den Daten befassen möchten, können Sie per FTP auf die im Dateisystem gespeicherten Protokolle zugreifen. Die FTP-Adressen werden im Abschnitt „Download Logs“ (Download-Log-Dateien) oder im Fenster „Overview“ (Übersicht) der Web App angezeigt. Sie denken vielleicht: „FTP ist für den Erhalt von Diagnose-Log-Dateien kompliziert. Gibt es nicht einen einfacheren Weg?“ Aber natürlich gibt es den! Im Azure Portal, genau dort, wo Sie Ihre Diagnose-Log-Dateien konfiguriert haben, finden Sie eine Log-Stream-Option. Können Sie erraten, was diese macht? Ich gebe Ihnen einen Tipp – es hat etwas mit dem Streaming Ihrer Log-Dateien zu tun.

Wenn Sie diese Schaltfläche im Azure Portal wählen, können Sie zwischen Anwendungs- und Webserver-Log-Dateien wählen. Diese Log-Dateien lesen aus den gleichen Diagnose-Log-Dateien, die in die Datei geschrieben werden. Es kann einige Minuten dauern, bis die Log-Datein im Stream angezeigt werden. Das Angezeigte hängt von den von Ihnen angegebenen Log-Datei-Ebenen ab sowie davon, ob Ihre Webanwendung irgendwelche Anwendungsereignisse generiert. Für die grundlegende HTML-Site ist der Stream eher uninteressant, aber er ist eine tolle Funktion für den Webbrowser. Abbildung 3.7 zeigt Beispiele des Streamings von Webserver-Log-Dateien im Azure Portal.

Jetzt testen

Sehen Sie sich die Streaming-Log-Dateien im Azure Portal an. Möglicherweise müssen Sie die Seite in Ihrem Webbrowser einige Male aktualisieren, um Aktivität in den Log-Dateien zu generieren.

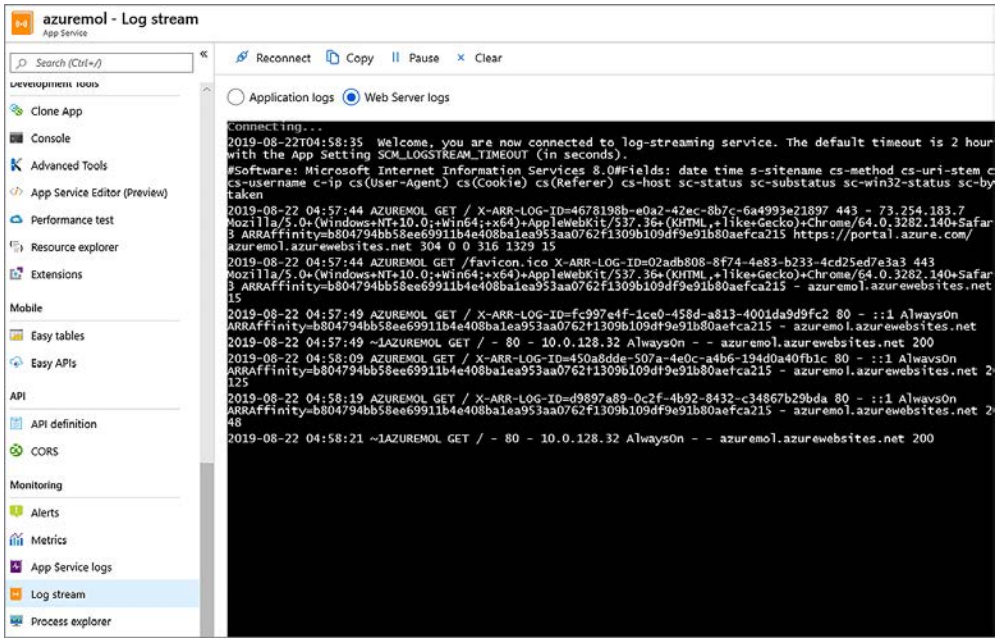


Abbildung 3.7 Sie können die Webserver-Log-Datei-Streams von Web Apps über Live-Log-Dateien aus Ihrer Anwendung anzeigen, um die Leistung der Anwendung zu überprüfen und zu debuggen. Das Konsolenfeld auf der rechten Seite des Bildschirms zeigt die Echtzeit-Streaming-Log-Dateien aus Ihrer Web App an.

Während Sie sich an Azure gewöhnen und die Azure CLI oder das Azure PowerShell-Modul verwenden, können Sie auch Log-Dateien mit diesen Tools streamen. Entwickler*innen können auch Remote-Debugging mit Visual Studio aktivieren oder Application Insights einrichten, damit Ihre Webanwendung Telemetrie für zusätzliche Dienste zur Überwachung und Diagnose bereitstellen kann. Das hauptsächliche Fazit hierbei ist, dass Sie beim Wechsel zu PaaS-Lösungen wie Web Apps weiterhin wichtige Diagnose-Log-Dateien und Anwendungsdaten zur Fehlerbehebung und Überwachung der Leistung Ihrer Webanwendungen abrufen.

3.4 Übung: Erstellen und Nutzen eines Bereitstellungsslots

Wir haben bereits besprochen, wie eine einfache HTML-Site erstellt und die Seite per Push mit Git an Azure Web Apps weitergeleitet wird. Was ist, wenn Sie jetzt einige neue Pizzen hinzufügen und diese ansehen möchten, bevor Sie die Website veröffentlichen, damit Kund*innen bestellen können? Finden wir heraus, wie ein Bereitstellungsslot verwendet wird, um einen Ort zum Hochladen Ihrer Änderungen bereitzustellen, sie dort zu überprüfen und dann an die Produktion zu übergeben:

1. Wählen Sie in Ihrer Web App „Deployment Slots“ (Bereitstellungsslots). Fügen Sie einen Bereitstellungsslot mit der Bezeichnung „Dev“ hinzu, klonen Sie jedoch keine Einstellungen aus dem vorhandenen Bereitstellungsslot.
2. Wenn Sie bereit sind, wählen Sie den Staging-Slot aus der Liste aus. Das Portal zeigt dieselben Konfigurations- und Protokollierungsoptionen wie der Produktionsslot, der zeigt, wie Sie die Einstellungen in diesem Bereitstellungsslot ändern können, ohne die Livewebsite zu beeinflussen.

3 Erkunden Sie dieses Mal die Optionen im Azure Portal für das Deployment Center. Sie sollten das lokale Git für die Quellcodeverwaltung verwenden, das den Build-Dienst von App Service für den Stagingslot nutzt. Dies geschah hinter den Kulissen, als Sie in der vorherigen Übung die Azure CLI verwendet haben. Sie haben jedoch auch andere Optionen in Bezug auf den Speicherort, von dem aus Sie Ihren Code bereitstellen können, und den Dienst, der diese Bereitstellung erstellt und erzeugt.

4 Wenn Sie fertig sind, kopieren Sie die geklonte Git-URI, wie https://azuremol-dev.scm.azurewebsites.net:443/azuremol.git. Beachten Sie, dass das Git-Repository den `-dev` für den Staging-Slot enthält.

Eine Beispielentwicklungsseite ist in den GitHub-Beispielen enthalten, die Sie zuvor geklont haben.

5 Wechseln Sie in der Azure Cloud Shell wie folgt zum Entwicklungsverzeichnis:

```
cd ~/azure-mol-samples-2nd-ed/03/dev
```

6 Initialisieren Sie wie zuvor Ihre Änderungen in Git, fügen Sie sie hinzu, und übertragen Sie sie mit den folgenden Befehlen:

```
git init && git add . && git commit -m "Pizza"
```

7 Erstellen Sie in Ihrem Stagingslot erneut einen Link zum neuen Git-Repository mit dem Befehl `git remote add dev`, , gefolgt von Ihrer Git-Bereitstellungs-URL des Stagingslots.

8 Verwenden Sie `git push dev master`, um Ihre Änderungen per Push in den Bereitstellungsslot zu verschieben.

9 Wählen Sie aus dem Übersichtsfenster im Azure Portal die URL zu Ihrem Staging-Slot. Die Änderung ist zwar nicht besonders groß, aber der Seitentitel lässt erkennen, dass es sich um die Entwicklungsversion handelt.

10 Was denken Sie, was im Azure Portal für die Web App geschieht, wenn Sie die Schaltfläche „Swap“ (Wechseln) auswählen, wie in Abbildung 3.8 gezeigt? Versuchen Sie es, und aktualisieren Sie dann die Hauptseite, wie https://azure-mol.azurewebsites.net, im Webbrowser.

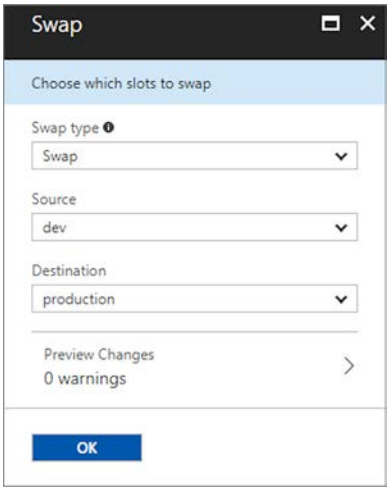


Abbildung 3.8 Wenn Sie die Slots für eine Web App wechseln, wählen Sie die Quell- und Zielinstanzen aus, die Sie ändern möchten. Sie können auch eine Vorschau des neuen Erscheinungsbilds anzeigen, bevor Sie die Änderungen in einer Live-Umgebung vornehmen.

Bereitstellungsslots, hinter den Kulissen

Wenn Sie Slots wechseln, befindet sich das, was im *Produktionsslot* öffentlich war, im *Entwicklungsslot*, und was sich im *Entwicklungsslot* befand, ist jetzt in der *Produktion* veröffentlicht. Nicht alle Einstellungen können gewechselt werden, z. B. SSL-Einstellungen und benutzungsdefinierte Domänen, doch in den meisten Fällen sind Bereitstellungsslots eine gute Möglichkeit zum Staging und Validieren von Inhalten, bevor diese für Ihre Kund*innen veröffentlicht werden. Sie können auch einen Wechsel mit *Vorschau* durchführen, mit dem Sie sicherstellen können, dass der gewechselte Inhalt korrekt funktioniert, bevor er für die Produktion veröffentlicht wird.

Für den Produktionseinsatz in DevOps-Workflows können Sie auch den automatischen Wechsel konfigurieren. Wenn hier ein Code-Commit in der Quellcodeverwaltung wie GitHub vermerkt wird, kann dadurch ein Build zu einem Azure Web Apps-Bereitstellungsslot ausgelöst werden. Sobald dieser Build fertiggestellt und die App bereit ist, Inhalte bereitzustellen, werden die Bereitstellungsslots automatisch gewechselt, um den Code für die Produktion zu veröffentlichen. Dieser Workflow wird in der Regel in einer Testumgebung verwendet, um zuerst die Codeänderungen zu überprüfen, nicht um direkt für die Produktion zu veröffentlichen.

4 Einführung zu Azure Storage

Einer Sache können wir uns in der IT-Welt sicher sein: Wenn etwas schief läuft, sind zwangsläufig Storage und Vernetzung schuld. Ich sage das als jemand, der in einem früheren Leben den Hut eines SAN-Admins getragen hat. Das Netzwerkteam war mein bester Freund. Das ist größtenteils ein Witz (dass wir beste Freunde waren), doch es spielt keine Rolle, wie gut eine Anwendung entwickelt und geschrieben ist: Die grundlegenden Infrastrukturstücke müssen am richtigen Fleck sein, um sie zu unterstützen. Halten Sie durch, in den kommenden Kapiteln entdecken wir Azure Storage und Azure Networking. Bei diesen Diensten neigen Sie vielleicht dazu, sie nur zu überfliegen, um mit den coolen darauffolgenden Kapiteln fortzufahren, aber es bringt einen deutlichen Mehrwert, einige Zeit zu investieren, um diese Kerndienste kennenzulernen. Die Pizza schmeckt dadurch zwar nicht besser, aber es kann Ihren Kund*innen die Bestellung von leckeren Pizzen vereinfachen.

Dieses Kapitel befasst sich mit den verschiedenen Storage-Typen von Azure und ihrer Verwendung. Wir sprechen auch über Redundanz- und Replikationsoptionen für den Azure Storage-Dienst und wie Sie die beste Leistung für Ihre Anwendungen erzielen können.

4.1 Verwaltete Datenträger

Vor einigen Jahren war Server-Storage teuer, langsam und zu kompliziert. Es war nicht ungewöhnlich, dass Storage-Anbieter Ihnen Hardware verkauften, die Hunderttausende Euro kostete und Tage oder sogar Wochen brauchte, bis eine Armee von Berater*innen und Ingenieur*innen sie konfigurierte. Als die Virtualisierung begann, sich in Rechenzentren zu etablieren und VMware und Hyper-V immer mehr akzeptiert wurden, kam es beim Speicher oft zu einem Engpass. Ganz zu schweigen von Firmware-Inkompatibilitäten zwischen Storage-Adaptern auf dem Server und auf dem Storage-Array, redundanten Netzwerkpfaden,

die hinten und vorne ausfallen, und SSDs, die als der einzige Weg zur Leistungssteigerung galten.

Hat Azure all diese Storage-Probleme auf magische Weise behoben? Natürlich nicht. Aber es abstrahiert 95 % dieser Sorgen, und Sie können sich darauf konzentrieren, tolle Erfahrungen für Ihre Kund*innen zu entwickeln und bereitzustellen. In diesem Kapitel geht es um die letzten 5 %, die Sie berücksichtigen müssen.

Der Azure Managed Disks-Dienst vereinfacht den Ansatz für den VM-Speicher. Verwaltete Datenträger abstrahieren viel der hinter den Kulissen stattfindenden Arbeit, um Ihnen ... nun, einen Datenträger bereitzustellen. Das ist alles, was Sie bei VMs wirklich beachten müssen: Wie groß und wie schnell sind sie, und womit sind sie verbunden? In diesem Buch und in allen Ihren eigenen Bereitstellungen in der Praxis sollten Sie immer verwaltete Datenträger für VMs verwenden. Verwaltete Datenträger sind die Standardoption, und es gibt nicht viele gute Gründe, dieses Verhalten zu ändern.

Vor der Zeit verwalteter Datenträger mussten Sie ein Storage-Konto mit einem einmaligen Namen erstellen, die Anzahl der virtuellen Datenträger einschränken, die Sie in jedem Konto erstellen, und manuell durch benutzungsdefinierte Datenträgerimages navigieren, um VMs in verschiedenen Regionen zu erstellen. Diese Arten von Datenträgern werden als *nicht verwaltete Datenträger* oder *klassische Datenträger* bezeichnet. Durch den Managed Disks-Dienst entfällt die Notwendigkeit eines Storage-Kontos. Sie werden auf „nur“ 50.000 Datenträger je Abonnement eingeschränkt und können VMs aus einem benutzungsdefinierten Image über mehrere Regionen hinweg erstellen. Sie erhalten außerdem die Möglichkeit, Momentaufnahmen von Datenträgern zu erstellen und zu verwenden, automatisch Daten im Ruhezustand zu verschlüsseln und Datenträger mit einer Größe von bis zu 64 TiB zu verwenden.

Warum ist das wichtig? Wenn Sie alte Dokumentation oder Blogbeiträge durchgehen, müssen Sie eventuell ein Storage-Konto für Ihre VMs erstellen. Stopp! Ja, Sie können VMs von nicht verwalteten auf verwaltete Datenträger konvertieren, aber bei einem Neuanfang sollten Sie versuchen, jedes Projekt von Anfang an mit verwalteten Datenträgern zu starten. Der Anwendungsfall für nicht verwaltete Datenträger ist eher, Abwärtskompatibilität mit bestehenden Bereitstellungen beizubehalten, obwohl ich empfehle, diese Workloads auf verwaltete Datenträger zu konvertieren.

4.1.1 Betriebssystemdatenträger

Erinnern Sie sich an die vorherige Zeile dazu, dass Sie SSDs kaufen müssen, wenn Sie die beste Leistung erzielen möchten? Leider gibt es in Azure tatsächlich keine magische Lösung, um diese Anforderung zu umgehen. Tut mir Leid. Denn SSDs übertreffen reguläre rotierende Festplatten deutlich. Es gibt physische Begrenzungen, wie schnell diese rotierenden Festplatten ... nun … rotieren. Die Ingenieur*innen von Microsoft konnten die Gesetze der Physik bisher noch nicht besiegen. Es gibt immer noch Anwendungsfälle für reguläre rotierende Festplatten, wie geringe Archivierungskosten, und wie bei herkömmlichen Storage-Arrays kann durch neueste Technologie mit einem Pool an rotierenden Festplatten gute Leistung erzielt werden.

Die erste und wichtigste Entscheidung, die Sie bei einer Azure-VM treffen müssen, ist die Art von Speicher, die verwendet werden soll:

- *Premium-SSD-Datenträger* – Datenträger: Verwenden Sie leistungsstarke SSD-Datenträger für optimale Leistung, mehr IOPS und niedrige Latenz; empfohlener Speichertyp für die meisten Produktionsworkloads.
- *Standard-SSDs* – Datenträger: Verwenden Sie Standard-SSDs und bieten Sie eine konstante Leistung im Vergleich zu Festplatten (HDDs). Diese Datenträger sind

ideal für die Entwicklung und für Tests von Workloads geeignet – oder für die Produktion mit knappem Budget und geringer Nachfrage.

- *Standard-HDDs* – verwenden herkömmlicher rotierender Festplatten für unregelmäßigen Datenzugriff wie für Archive oder Sicherungen.

Die von Ihnen gewählte VM-Größe hilft zu bestimmen, welche Art von Speicher Sie auswählen können. In Kapitel 2 haben Sie beim Erstellen einer VM eine Größe ausgewählt, um schnell eine VM bereitzustellen. Die Standardeinstellung war wahrscheinlich so etwas wie eine VM der Serie D2S_v3, mit der Sie Zugriff auf hochwertige SSDs hatten. Woran erkennen Sie, mit welchen VMs Sie auf Premium-SSDs zugreifen können? Suchen Sie in der VM-Größe nach einem *s*, das für *SSD steht*. Es gibt zwar ein paar Ausnahmen von der Regel, aber das ist ein gutes Muster. Anhand der folgenden Beispiele können Sie feststellen, welche VMs auf Premium-Datenträger und welche VMs auf Standard-SSDs oder HDDs zugreifen können:

- Mit VMs der Serie D2S_v3, F*s*, GS und L*s* erhalten Sie Zugriff auf Premium-SSDs.
- VMs der Serie D, A, F und M bieten nur Zugriff auf Standard-SSD oder HDDs.

Wenn Sie eine VM-Größe wählen, mit der Sie Premium-SSD-Platten verwenden können, besteht allerdings keine Verpflichtung dazu. Sie können Standard-SSDs oder HDDs erstellen und verwenden. Durch die Wahl von Premium-SSDs machen Sie die Anwendung zukunftssicher und erhalten die Möglichkeit, Hochleistungs-SSDs nach Bedarf zu verwenden, ohne die Größe Ihrer VMs ändern zu müssen und dabei geringe Ausfallzeiten zu verursachen. Alle VM-Größen können Standard-SSDs verwenden.

Kurzlebiger Betriebssystemdatenträger

Es gibt eine spezielle Art von Betriebssystemdatenträger, die als *kurzlebiger Datenträger* bezeichnet wird. Es handelt sich immer noch um eine Art verwalteten Datenträger, aber er befindet sich lokal auf dem zugrunde liegenden Azure-Host. Diese Tatsache macht einen kurzlebigen Datenträger wirklich schnell, mit geringer Latenz.

Da die Daten nicht auf ein Remote Storage-Array geschrieben werden, bleiben sie bei einem VM-Neustart möglicherweise nicht erhalten, wenn Sie auf einen anderen zugrunde liegenden Host wechseln. Kurzlebige Datenträger eignen sich hervorragend für zustandslose Workloads, die jedes Mal mit einem neuen Image booten können und keine Daten lokal speichern müssen, auf die sie bei jedem Neustart zugreifen können

Nur bestimmte VM-Größen unterstützen kurzlebige Datenträger. Es entstehen aber keine zusätzlichen Kosten für ihre Verwendung, und sie sind in allen Regionen verfügbar. Sie verlieren einige Funktionen für Dinge wie Azure Site Recovery und Azure Disk Encryption (Kapitel 13 bzw. 14), wenn Sie allerdings Speicher mit hoher Geschwindigkeit und niedriger Latenz benötigen, sollten Sie kurzlebige Datenträger in Betracht ziehen.

Jetzt testen

Woran erkennen Sie, welche VM-Größen Ihnen zur Verfügung stehen? Öffnen Sie die Cloud Shell im Azure Portal. Geben Sie folgenden Befehl ein (verwenden Sie Ihre eigene Region):

```
az vm list-sizes --location eastus --output table
```

Denken Sie daran: Jede Größe mit einem „s“ bietet Ihnen Zugriff auf Premium-SSDs.

4.1.2 Temporäre Festplatten und Datenträger

Nun haben Sie herausgefunden, welches Leistungsniveau Sie für Ihre Anwendungen benötigen, kommen wir also zu den nächsten Puzzleteilen. Datenträger sind auf zwei Arten verbunden:

- *Temporäre Datenträger* – jeder VM ist automatisch ein lokaler SSD-Speicher des zugrunde liegenden Hosts angehängt, der eine kleine Menge Hochleistungsspeicher bietet. Achten Sie gut darauf, wie Sie diesen temporären Datenträger verwenden. Wie der Name schon sagt, kann es sein, dass diese Platte nicht mit der VM erhalten bleibt. Wenn die VM bei einem Wartungsereignis in einen neuen Host wechselt, wird eine neue temporäre Festplatte angehängt. Daten, die Sie dort gespeichert haben, gehen verloren. Der temporäre Datenspeicher dient speziell als Scratch-Speicher oder Anwendungscache.
- *Datenträger* – alle Datenträger, die Sie speziell erstellen und an die VM anhängen, verhalten sich in Bezug auf Partitionen, Dateisysteme und persistente Bereitstellungspunkte wie erwartet. Die Datenträger werden neu angehängt, da sich die VM im Azure-Rechenzentrum umherbewegt – sie sind der Ort, an dem der Großteil Ihrer Anwendungen und Daten gespeichert werden sollte. Sie können auch weiterhin Speicherplätze oder Software-RAID verwenden, um Datenträger auf VM-Ebene zu bündeln, um eine noch höhere Leistung zu erzielen.

Es gibt einen speziellen Datenträgertyp, den Sie an eine VM anschließen können, wenn Sie maximale Leistung und niedrige Latenz benötigen: Ultra Disks. Diese Datenträger liegen eine Stufe über Premium-SSDs und sind nur für Datenträger verfügbar. Ultra Disks sind für große Datenbanken und datenintensive Workloads wie SAP HANA konzipiert. Von welcher Geschwindigkeit sprechen wir? Zum Zeitpunkt der Erstellung dieses Artikels können Ultra Disks bis zu 64 TiB groß sein und bieten bis zu 160.000 IOPS pro Datenträger mit einem maximalen Durchsatz von 2.000 MBit/s.

4.1.3 Datenträger-Caching-Optionen

Es ist auch wichtig, den Datenträger des Betriebssystems zu bedenken, der bei der VM enthalten ist. Wenn Sie eine VM erstellen, erhalten Sie immer mindestens einen Datenträger: den Datenträger, auf dem das Betriebssystem selbst installiert ist. Es ist verlockend, diesen Datenträger zu verwenden, um Ihre Anwendungen zu installieren oder Log-Dateien darauf zu schreiben. Außer, wenn Sie eine kleine Proof-of-Concept-Bereitstellung ausführen, führen Sie Ihre Anwendungen nicht auf dem Datenträger des Betriebssystems aus. Es ist ziemlich wahrscheinlich, dass Sie nicht die Leistung erzielen, die Sie wünschen.

Für Datenträger auf Azure können Caching-Richtlinien gelten. Standardmäßig ist für den Datenträger des Betriebssystems *Lese-/Schreib*-Caching aktiviert. Diese Art von Caching ist in der Regel nicht ideal für Anwendungsworkloads, die beispielsweise Log-Dateien oder Datenbanken schreiben. Für Datenträger hingegen ist die Standard-Caching-Richtlinie *keine*. Dies ist eine gute Richtlinie für Workloads, die eine Menge von Schreibvorgängen durchführen. Sie können auch eine *Lese*-Caching-Richtlinie verwenden, die sich besser für Application-Workloads eignet, die in erster Linie Daten von den Datenträgern lesen.

In der Regel können Sie immer Datenträger anhängen und verwenden, um Ihre Anwendungen zu installieren und auszuführen. Selbst die Standard-Caching-Richtlinie „keine“ bietet wahrscheinlich eine bessere Leistung als die Lese-/Schreib-Caching-Richtlinie des Datenträgers des Betriebssystems.

4.2 Hinzufügen von Datenträgern zu einer VM

In diesem Abschnitt erfahren Sie, wie Sie einer VM während der Erstellung Datenträger hinzufügen. In Kapitel 2 haben Sie eine VM mit dem Azure Portal erstellt. Dieses Mal verwenden wir die Azure CLI, um eine VM zu erstellen. Die Azure CLI bietet eine schnelle Möglichkeit, eine VM zu erstellen und gleichzeitig einen Datenträger anzuhängen.

Jetzt testen

Um eine VM zu erstellen und Datenträger in Aktion zu erleben, führen Sie die folgenden Schritte aus:

1. Erstellen Sie in der Azure Cloud Shell mit dem Befehl `az group create`, und geben Sie einen Namen für die Ressourcengruppe an, zusammen mit einem Standort:

   ```
   az group create --name azuremolchapter4 --location eastus
   ```

2. Erstellen Sie mit dem Befehl `az vm create` eine VM. Mit dem letzten Parameter, `--data-disk-sizes-gb`, können Sie zusammen mit der VM einen Datenträger erstellen. In der Übung am Ende des Kapitels können Sie eine Verbindung zu dieser VM herstellen und die Datenträger initialisieren.
   - Sie können für diese Übung eine Linux- oder Windows-VM erstellen. Wenn Sie sich mit Linux auskennen oder lernen möchten, wie eine Festplatte für Linux initialisiert und vorbereitet wird, verwenden Sie den folgenden Befehl, um eine Ubuntu LTS-VM zu erstellen:

     ```
     az vm create \
       --resource-group azuremolchapter4 \
       --name storagevm \
       --image UbuntuLTS \
       --size Standard_B1ms \
       --admin-username azuremol \
       --generate-ssh-keys \
       --data-disk-sizes-gb 64
     ```

   - Wenn Sie Windows bevorzugen, verwenden Sie den folgenden Befehl, um eine Windows Server 2019-VM zu erstellen. Sie können dann per RDP eine Verbindung mit der VM herstellen, um die Datenträger später zu konfigurieren:

     ```
     az vm create \
       --resource-group azuremolchapter4 \
       --name storagevm \
       --image Win2019Datacenter \
       --size Standard_B1ms \
       --admin-username azuremol \
       --admin-password P@ssw0rd! \
       --data-disk-sizes-gb 64
     ```

   - Es dauert einige Minuten, um die VM zu erstellen. Der VM ist bereits ein Datenträger angehängt und sie ist einsatzbereit.

Was ist, wenn Sie nach einigen Wochen oder Monaten einen weiteren Datenträger hinzufügen möchten? Verwenden wir die Azure CLI erneut, um zu sehen, wie schnell wir einen Datenträger hinzufügen können. Der Prozess ist für Linux- und Windows-VMs identisch. Sie müssen Azure nur mitteilen, dass ein neuer Datenträger erstellt und dieser an Ihre VM angehängt werden soll.

Jetzt testen

Fügen Sie wie im Folgenden gezeigt einen weiteren Datenträger zu Ihrer VM hinzu.

Erstellen Sie einen neuen Datenträger mit dem Befehl `az vm disk attach`. Geben Sie einen Namen und eine Größe für den Datenträger an. Erinnern Sie sich an die vorherige Diskussion über Standard- und Premium-Datenträger? Im folgenden Beispiel erstellen Sie eine Premium-SDD:

```
az vm disk attach \
    --resource-group azuremolchapter4 \
    --vm-name storagevm \
    --name datadisk \
    --size-gb 64 \
    --sku Premium_LRS \
    --new
```

Erkennen Sie den letzten Teil dieses Speichertyps? *LRS* steht für *lokal redundanten Speicher*. Wir sehen uns die Redundanzoptionen in Abschnitt 4.3.3 an.

Mit nur zwei Befehlen haben Sie eine VM mit der Azure CLI erstellt, die einen Datenträger enthält, und dann simuliert, wie später ein weiterer Datenträger angefügt wird. Nur weil Sie diese Datenträger angehängt haben, bedeutet das nicht, dass Sie direkt Daten darauf schreiben können. Wie bei jedem Datenträger, sei es physisch mit einem On-Premises-Server oder virtuell mit einer VM verbunden, müssen Sie den Datenträger initialisieren und eine Partition sowie ein Dateisystem erstellen. Sie können dies in der optionalen Übung am Ende des Kapitels tun.

4.3 Azure Storage

Storage ist vielleicht nicht gerade das offensichtlichste Thema bei der Entwicklung und Ausführung von Anwendungen, aber es ist eine breite Dienstleistung, die viel mehr abdeckt, als Sie denken. Der Azure Storage-Dienst bietet viel mehr als nur einen Ort zum Speichern von Dateien oder virtuellen Datenträgern für Ihre VMs.

Sehen wir uns an, was Ihr fiktives Pizzaunternehmen benötigt, um eine App zur Verarbeitung von Bestellungen von Kund*innen zum Mitnehmen oder zur Lieferung zu entwickeln. Die App benötigt einen Datenspeicher, der die verfügbaren Pizzen, Beläge und Preise enthält. Beim Empfang und der Verarbeitung von Bestellungen muss die App über eine Möglichkeit verfügen, Nachrichten zwischen den verschiedenen Anwendungskomponenten zu senden. Für die Frontend-Website sind dann noch appetitliche Bilder notwendig, damit den Kund*innen beim Anblick der Pizzen das Wasser im Mund zusammenläuft. Wie Sie in Abbildung 4.1 sehen können, verfügt Azure Storage über eine Vielzahl von Speicherfunktionen und kann alle drei dieser Anforderungen abdecken.

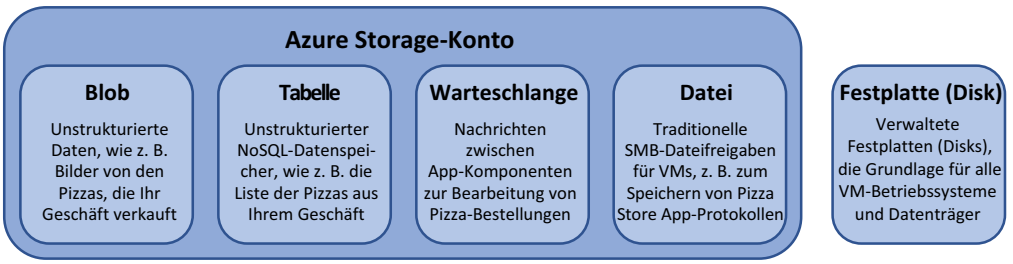


Abbildung 4.1 Ein Azure Storage-Konto ermöglicht es Ihnen, eine Vielzahl von Speicherfunktionen zu erstellen und zu nutzen, weit über den Speicherort von Dateien hinaus.

- *Blog Storage* – für unstrukturierte Daten wie Mediendateien und Dokumente. Anwendungen können Daten, zum Beispiel Bilder, im Blob Storage aufbewahren und dann rendern. Sie können Bilder Ihrer Pizzen im Blob Storage ablegen.
- *Table Storage* – für unstrukturierte Daten in einem NoSQL-Datenspeicher. Planen Sie wie bei jeder Debatte über SQL- versus NoSQL-Datenspeicher Ihre Anwendung, und schätzen Sie die Leistungsanforderungen, wenn es um die Verarbeitung großer Datenmengen geht. Sie können die Liste der Pizzen Ihrer Speisekarte im Table Storage ablegen. In Abschnitt 4.3.1 wird NoSQL im Detail erkundet.
- *Queue Storage* – für die zuverlässige und konsistente Kommunikation für Cloud-Anwendungen zwischen verschiedenen Ebenen und Komponenten. Sie können Nachrichten, die zwischen Anwendungskomponenten gesendet werden, erstellen, lesen und löschen. Sie könnten Warteschlangenspeicher verwenden, um Nachrichten zwischen dem Web-Front-End, wenn Kund*innen eine Bestellung aufgeben, und dem Backend zu übertragen, um die Bestellung zu verarbeiten und Pizzen zu backen.
- *File Storage* – für eine gute altmodische SMB-Dateifreigabe (Server Message Block), die sowohl über Windows- als auch über Linux-/macOS-Plattformen zugänglich ist. Oft zur Zentralisierung der Log-Datei-Erfassung von VMs verwendet.

Der Azure Storage für VMs ist eine unkomplizierte Angelegenheit. Sie erstellen und verwenden durch Azure Managed Disks, eine Art virtuelle Festplatte (VHD), die einen großen Teil der Entwurfsüberlegungen bezüglich Leistung und Bereitstellung der virtuellen Datenträger der gesamten Plattform abstrahiert. Sie erstellen eine VM, fügen verwaltete Datenträger hinzu und überlassen es der Azure-Plattform, die Redundanz und Verfügbarkeit auszuklügeln.

4.3.1 Table Storage

Besprechen wir einige Arten von Datenspeichern. Zunächst der *Table Storage*. Die meisten kennen wahrscheinlich traditionelle SQL-Datenbanken wie Microsoft SQL Server, MySQL oder PostgreSQL. Dabei handelt es sich um *relationale Datenbanken*, die aus mindestens einer Tabelle bestehen, die mindestens einen Datensatz enthält. Relationale Datenbanken werden häufig für die Anwendungsentwicklung verwendet und können auf strukturierte Weise entwickelt, visualisiert und abgefragt werden – dafür steht das *S* in *SQL* (für Structured Query Language).

NoSQL-Datenbanken sind ein wenig anders. Sie folgen nicht dem gleichen strukturierten Ansatz und Daten werden nicht in Tabellen gespeichert, in denen jede Zeile die gleichen Felder enthält. Es gibt verschiedene Implementierungen von NoSQL-Datenbanken: Beispiele sind MongoDB und CouchDB. Die angepriesenen Vorteile von NoSQL-Datenbanken sind, dass sie horizontal skalieren (was bedeutet, dass Sie mehr Server hinzufügen können, anstatt mehr Speicher oder CPUs hinzuzufügen), größere Datenmengen verarbeiten können und bei der Verarbeitung dieser großen Datensätze effizienter sind.

Wie Daten in einer NoSQL-Datenbank gespeichert werden, lässt sich in einigen Kategorien definieren:

- *Schlüssel/Wert*, z. B. Redis
- *Spalte*, z. B. Cassandra
- *Dokument*, z. B. MongoDB

Jeder Ansatz hat Vor- und Nachteile bezüglich Leistung, Flexibilität und Komplexität. Eine Azure-Speichertabelle verwendet Schlüssel/Wert-Speicher und ist gut für den Einstieg in NoSQL-Datenbanken geeignet, wenn Sie SQL-Datenbanken wie Microsoft SQL oder MySQL gewöhnt sind.

Sie können den Azure Storage Explorer von https://azure.microsoft.com/features/storage-explorer herunterladen und installieren, wenn Sie die Daten visualisieren möchten. Das müssen Sie aber nicht jetzt tun. Storage Explorer ist ein großartiges Tool, um herauszufinden, wie Tabellen und Warteschlangen in Aktion aussehen. In diesem Kapitel möchte ich Sie nicht zu tief in NoSQL-Datenbanken führen – in Kapitel 10 erkunden wir einige coole NoSQL-Datenbanken mit Azure Cosmos DB weiter. Tatsächlich verwenden Sie in der folgenden Übung die Cosmos DB-API, um eine Verbindung zu Azure Storage herzustellen und eine Tabelle zu erstellen. Die Verwendung von Azure-Tabellen ist eher eine Einführung in NoSQL-Datenbanken als ein solides Beispiel für die Nutzung in der Produktion.

Lassen Sie uns vorerst schnell eine Beispielapp ausführen, um zu sehen, wie Sie Daten hinzufügen und abfragen können, so wie Sie es mit der tatsächlichen Anwendung tun würden. Diese Beispiele sind einfach, aber sie zeigen, wie Sie die Pizzen, die Sie anbieten, und die einzelnen Preise speichern können. Anstatt etwas Großes wie Microsoft SQL Server oder MySQL zu verwenden, sollten wir eine NoSQL-Datenbank mit Azure-Tabellenspeicher verwenden.

Jetzt testen

Führen Sie die folgenden Schritte aus, um Azure-Tabellen in Aktion zu erleben:

1. Öffnen Sie das Azure Portal in einem Webbrowser und dann Cloud Shell.
2. In Kapitel 3 erhielten Sie eine Kopie der Azure-Beispiele von GitHub. Wenn Sie das nicht getan haben, sichern Sie sich eine Kopie wie folgt:

   ```
   git clone https://github.com/fouldsy/azure-mol-samples-2nd-ed.git
   ```

3. Wechseln Sie zu dem Verzeichnis, das die Azure Storage-Beispiele enthält:

   ```
   cd ~/azure-mol-samples-2nd-ed/04
   ```

4. Installieren Sie einige Python-Abhängigkeiten, wenn sie nicht bereits installiert sind. Hier installieren Sie das `azurerm`-Paket, das die Kommunikation verwaltet, durch die Sie Azure-Ressourcen erstellen und verwalten können, sowie zwei `azure`-Pakete, die die zugrunde liegenden Python-SDKs für Azure Cosmos DB und Storage sind:

   ```
   pip install --user azurerm azure-cosmosdb-table azure-storage-
   ➥queue==2.1.0
   ```

   Was bedeutet `--user` beim Installieren der Pakete? Wenn Sie Azure Cloud Shell verwenden, können Sie keine Pakete im Kernsystem installieren. Sie haben dazu keine Berechtigung. Stattdessen werden die Pakete in der Umgebung Ihres Benutzenden installiert. Diese Paketinstallationen bestehen über Sessions hinweg, und Sie können all die tollen Azure SDKs in diesen Beispielen verwenden.

5. Führen Sie die Python-Beispielanwendung für Tabellen aus. Befolgen Sie die Aufforderungen, und genießen Sie Ihre Pizza:

   ```
   python storage_table_demo.py
   ```

Snakes on a Plane

Python ist eine weit verbreitete Programmiersprache, die oft in Einführungskursen für Informatik verwendet wird. Wenn Sie hauptsächlich im Bereich IT-Betrieb oder Administration tätig sind, können Sie Python als eine leistungsfähige Skriptsprache sehen, die über Betriebssysteme hinweg funktioniert. Python dient nicht nur zur Skripterstellung, sondern kann auch zur Entwicklung komplexer Anwendungen verwendet werden. Beispielsweise ist die Azure CLI, die Sie verwenden, in Python geschrieben.

Ich verwende Python für einige der Beispiele in diesem Buch, damit sie außerhalb der Cloud Shell ohne Modifikation funktionieren. macOS und Linux-Distributionen enthalten nativ Python. Windows-Benutzer*innen können Python herunterladen und schnell installieren, um diese Skripte lokal auszuführen. Python ist ideal für Benutzer*innen mit wenig bis gar keiner Programmiererfahrung geeignet sowie für erfahrenere Entwickler*innen, die mit anderen Sprachen vertraut sind. Die Azure-Dokumentation für Azure Storage und viele andere Dienste bietet Unterstützung für eine Reihe von Sprachen, darunter .NET, Java und Node.js. Sie sind nicht auf die Verwendung von Python beschränkt, wenn Sie Ihre eigenen Anwendungen entwickeln, die Tabellen verwenden.

Das Quick Python Book, 3. Ausgabe, von Naomi Ceder (http://mng.bz/6QZA), bringt Sie auf den neuesten Stand, wenn Sie mehr erfahren möchten. Es gibt auch einen videobasierten Kurs für *Get Programming with Python in Motion*, von Ana Bell (http://mng.bz/oPap).

4.3.2 Queue Storage

Azure Tables sind cool, wenn Sie Ihre Füße zum ersten Mal in das Meer der Anwendungsentwicklung für die Cloud tauchen. Wenn Sie damit beginnen, Anwendungen nativ in der Cloud zu entwickeln und zu verwalten, unterteilen Sie eine Anwendung in der Regeln in kleinere Komponenten, die alle selbst skalieren und Daten verarbeiten können. Um diesen verschiedenen Komponenten die Kommunikation und das Senden von Daten zu ermöglichen, ist typischerweise eine Form von Nachrichtenwarteschlange erforderlich. Hier kommt Azure Queues ins Spiel.

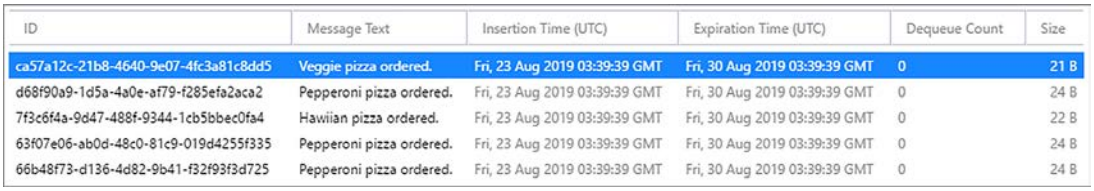
Der Azure Queues-Dienst ermöglichen es Ihnen, Nachrichten, die kleine Datenbrocken enthalten, zu erstellen, zu lesen und dann zu löschen. Diese Nachrichten werden von verschiedenen Anwendungskomponenten erstellt und abgerufen, während sie Daten hin und her senden. Azure Queues löscht Nachrichten erst, wenn durch eine Anwendungskomponente bestätigt wird, dass sie mit der Verarbeitung der Nachrichtendaten fertig ist.

Jetzt testen

Um Azure Queues in Aktion zu erleben, führen Sie das folgende Python-Skript aus demselben „azure-samples/4“-Verzeichnis aus. Befolgen Sie die Anweisungen, um zu sehen, wie Nachrichten aus der Warteschlange geschrieben, gelesen und gelöscht werden:

```
python storage_queue_demo.py
```

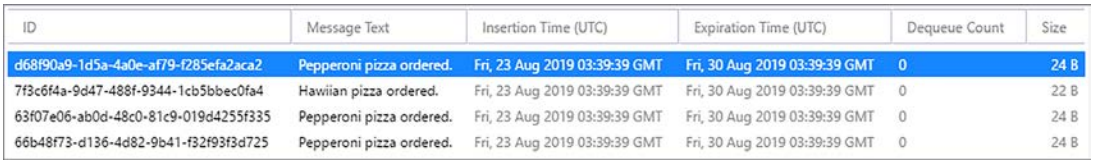
Fahren wir mir der Beispielanwendung fort, die Pizzabestellungen verarbeitet. Sie haben möglicherweise eine Frontend-Anwendungskomponente, mit der Kund*innen interagieren, um Pizzen zu bestellen, und dann eine Nachrichtenwarteschlange, die Nachrichten an eine Backend-Anwendungskomponente sendet, die diese Bestellungen verarbeitet. Wenn Bestellungen eingehen, können Nachrichten in der Warteschlange visualisiert werden (siehe Abbildung 4.2).



ID	Message Text	Insertion Time (UTC)	Expiration Time (UTC)	Dequeue Count	Size
ca57a12c-21b8-4640-9e07-4fc3a81c8dd5	Veggie pizza ordered.	Fri, 23 Aug 2019 03:39:39 GMT	Fri, 30 Aug 2019 03:39:39 GMT	0	21 B
d68f90a9-1d5a-4a0e-af79-f285efa2aca2	Pepperoni pizza ordered.	Fri, 23 Aug 2019 03:39:39 GMT	Fri, 30 Aug 2019 03:39:39 GMT	0	24 B
7f3c6f4a-9d47-488f-9344-1cb5bbec0fa4	Hawiian pizza ordered.	Fri, 23 Aug 2019 03:39:39 GMT	Fri, 30 Aug 2019 03:39:39 GMT	0	22 B
63f07e06-ab0d-48c0-81c9-019d4255f335	Pepperoni pizza ordered.	Fri, 23 Aug 2019 03:39:39 GMT	Fri, 30 Aug 2019 03:39:39 GMT	0	24 B
66b48f73-d136-4d82-9b41-f32f93f3d725	Pepperoni pizza ordered.	Fri, 23 Aug 2019 03:39:39 GMT	Fri, 30 Aug 2019 03:39:39 GMT	0	24 B

Abbildung 4.2 Nachrichten gehen bei der Frontend-Anwendungskomponente ein. Dabei wird in der Eigenschaft `Nachrichtentest` detailliert beschrieben, welche*r Kund*in welche Pizza bestellt hat.

Während die Backend-Anwendungskomponente die Pizzabestellung verarbeitet, werden die Nachrichten aus der Warteschlange entfernt. Abbildung 4.3 zeigt, wie die Warteschlange aussieht, wenn die Gemüsepizza im Ofen ist und diese erste Nachricht entfernt wird.



ID	Message Text	Insertion Time (UTC)	Expiration Time (UTC)	Dequeue Count	Size
d68f90a9-1d5a-4a0e-af79-f285efa2aca2	Pepperoni pizza ordered.	Fri, 23 Aug 2019 03:39:39 GMT	Fri, 30 Aug 2019 03:39:39 GMT	0	24 B
7f3c6f4a-9d47-488f-9344-1cb5bbec0fa4	Hawiian pizza ordered.	Fri, 23 Aug 2019 03:39:39 GMT	Fri, 30 Aug 2019 03:39:39 GMT	0	22 B
63f07e06-ab0d-48c0-81c9-019d4255f335	Pepperoni pizza ordered.	Fri, 23 Aug 2019 03:39:39 GMT	Fri, 30 Aug 2019 03:39:39 GMT	0	24 B
66b48f73-d136-4d82-9b41-f32f93f3d725	Pepperoni pizza ordered.	Fri, 23 Aug 2019 03:39:39 GMT	Fri, 30 Aug 2019 03:39:39 GMT	0	24 B

Abbildung 4.3 Beim Verarbeiten der einzelnen Nachrichten werden sie aus der Warteschlange entfernt. Die erste Nachricht in Abbildung 4.2 wurde entfernt, nachdem sie von der Backend-Anwendungskomponente verarbeitet wurde.

4.3.3 Storage-Verfügbarkeit und -Redundanz

Azure-Rechenzentren sind für redundante Internetanschlüssen, Stromerzeuger, mehrere Netzwerkpfade, Speicher-Arrays und so weiter fehlertolerant konzipiert. Sie müssen immer noch Ihren Teil zum Entwerfen und Ausführen von Anwendungen beitragen. Bei Azure Storage wählen Sie aus, welche Speicherredundanzebene Sie benötigen. Das variiert für jede Anwendung und je nachdem, wie wichtig die Daten sind. Prüfen wir die verfügbaren Speicherredundanzoptionen:

- *Lokal redundanter Speicher (LRS)* – Ihre Daten werden dreimal im einzelnen Rechenzentrum repliziert, in dem Ihr Storage-Konto erstellt wurde. Diese Option ermöglicht Redundanz im Falle eines Hardwareausfalls, aber wenn das gesamte Rechenzentrum ausfällt (selten, aber möglich), gehen Ihre Daten verloren.

- *Zonenredundanter Speicher (ZRS)* – Die nächste Ebene von LRS ist es, Ihre Daten dreimal in zwei oder drei Rechenzentren in einer Region (wenn es mehrere Rechenzentren in einer Region gibt) oder über mehrere Regionen hinweg zu replizieren. ZRS ist auch über Verfügbarkeitszonen hinweg verfügbar, die wir in Kapitel 7 näher untersuchen.
- *Georedundanter Storage (GRS)* – mit GRS werden Ihre Daten dreimal in der Primärregion repliziert, in der Ihr Speicher erstellt wurde, und dann dreimal in einer gekoppelten Region. Die gekoppelte Region liegt in der Regel Hunderte oder mehr Kilometer entfernt. Zum Beispiel ist der Westen der USA mit dem Osten der USA gekoppelt, Nordeuropa mit Westeuropa, Südostasien mit Ostasien. GRS bietet eine großartige Redundanzoption für Produktionsanwendungen.
- *Schreibgeschützter georedundanter Storage (Read-Access Geo-Redundant Storage, RA-GRS)* – dies ist die Premium-Datenredundanzoption. Ihre Daten werden über gekoppelte Regionen hinweg wie bei GRS repliziert, aber Sie erhalten auch schreibgeschützten Zugriff auf die Daten in der sekundären Zone.

4.4 Übung: Erkunden von Azure Storage

Hier haben Sie die Möglichkeit, Ihre Fähigkeiten zu testen. Wählen Sie eine der folgenden Aufgaben aus, die Sie für Ihre Übung erledigen möchten.

4.4.1 Fokus auf die VM

Wenn Sie sich bei einer VM anmelden und sehen möchten, dass der Prozess zum Initialisieren eines Datenträgers und Erstellen eines Dateisystems der gleiche ist wie für jede andere VM, mit der Sie gearbeitet haben, versuchen Sie eine dieser Übungen:

1. Melden Sie sich bei der VM an, die Sie in Abschnitt 4.2 erstellt haben. Je nach Wahl verbinden Sie sich mit SSH oder RDP.
2. Initialisieren Sie den Datenträger, und erstellen Sie eine Partition:
   - Unter Linux ist der Ablauf `fdisk`, `mkfs` und dann `mount`.
   - Unter Windows verwenden Sie die Sequenz, mit der sie vertraut sind – wahrscheinlich „Datenträgerverwaltung“ > „Initialisieren“ > „Volume erstellen“ > „Format“.

4.4.2 Fokus auf Entwickelnde

Wenn Sie eher Entwickler*in sind und nicht herausfinden möchten, wie Datenträger auf einer VM initialisiert werden, gehen Sie zurück zur Cloud Shell, und erkunden Sie die beiden Python-Demos, die Tabellen und Warteschlangen verwenden. Auch wenn Python neu für Sie ist, sollten Sie in der Lage sein, dem Geschehen zu folgen:

- Denken Sie an einige Szenarien, in denen Sie Tabellen oder Warteschlangen in Ihre eigenen Anwendungen implementieren können. Was ist beispielweise nötig, um eine Cloud Native-Anwendung mit einzelnen Anwendungskomponenten zu erstellen, die Warteschlangen nutzen könnten?
- Modifizieren Sie eines der Beispiele, das Sie interessiert, um ein zusätzliches Element in der Pizzakarte zu erstellen (wenn sie eine Tabelle ist) oder eine neue Pizzabestellnachricht (wenn es sich um eine Warteschlange handelt).

5 Grundlagen zu Azure Networking

In Kapitel 4 haben wir den Azure Storage-Dienst erkundet. Einer der anderen Kerndienste für Cloud-Anwendungen ist Azure Networking. Es gibt eine Menge leistungsstarker Netzwerkfunktionen bei Azure, um Ihren Datenverkehr im wirklich globalen Maßstab zu sichern und weiterzuleiten. Diese Funktionen sollen dabei helfen, sich darauf zu konzentrieren, wie Sie Ihre Apps entwickeln und pflegen können, sodass Sie sich keine Gedanken über Details wie IP-Adressen und Route-Tabellen machen müssen. Wenn Sie eine Website zur Verarbeitung von Pizzabestellungen entwickeln und bereitstellen, muss diese die Kundendaten sicher übermitteln und den Zahlungsverkehr sicher verarbeiten.

In diesem Kapitel untersuchen wir virtuelle Netzwerke und Subnetze von Azure, und sprechen darüber, wie virtuelle Netzwerkschnittstellen erstellt werden können. Um den Datenstrom zu sichern und zu steuern, erstellen Sie Netzwerksicherheitsgruppen und -regeln. Wenn Netzwerke für Sie neu sind oder wenn es eine Weile her ist, dass Sie mit IP-Adressen und Netzwerkkarten arbeiten mussten, kann es etwas länger dauern, dieses Kapitel durchzuarbeiten. Es gibt jede Menge „Jetzt testen"-Übungen. Es lohnt sich jedoch, sich die Zeit zu nehmen, um dieses Kapitel zu verstehen, da Netzwerke für viele der Azure-Dienste wichtig sind.

5.1 Virtuelle Netzwerkkomponenten

Denken Sie daran, wie viele Kabel sich hinter Ihrem Computertisch oder in Ihrem Entertainment Center befinden. Denken Sie nun an all die Kabel, die erforderlich sind, um die Computer in einem bestimmten Stockwerk eines Bürogebäudes zu verbinden. Wie sieht es mit dem gesamten Bürogebäude aus? Waren Sie schon einmal in einem Rechenzentrum oder haben Fotos von einem gesehen? Versuchen Sie sich vorzustellen, wie groß die Azure-Rechenzentren sind. Und stellen Sie sich nun Dutzende von Azure-Rechenzentren auf der ganzen Welt vor. Mathematik ist nicht

meine Stärke, also kann ich nicht berechnen, wie viele Tausende Kilometer von Netzwerkkabeln verwendet werden, um den gesamten Datenverkehr in Azure zu verwalten.

Die Netzwerkkonnektivität ist ein entscheidender Bestandteil des modernen Lebens. Bei Azure steht das Netzwerk im Mittelpunkt der gesamten Kommunikation. Für all die Tausenden von physischen Netzwerkgeräten und die kilometerlangen Netzwerkkabel, die alles in einem Azure-Rechenzentrum verbinden, arbeiten Sie mit *virtuellen* Netzwerkressourcen. Wie geht das? Softwaredefinierte Netzwerke. Wenn Sie eine VM oder eine Web App erstellen, muss ein(e) Techniker*in nicht durch das Azure-Rechenzentrum laufen, um Kabel für Sie physisch anzuschließen und IP-Adressen zuzuweisen (obwohl das lustig zu beobachten wäre). Stattdessen werden alle Ihre Netzwerkressourcen, die Ihre gesamte Netzwerkumgebung definieren, logisch von der Azure-Plattform verwaltet. Während Sie dieses Kapitel durcharbeiten, zeigt Abbildung 5.1 die virtuellen Netzwerkkomponenten, die Sie entwickeln werden.

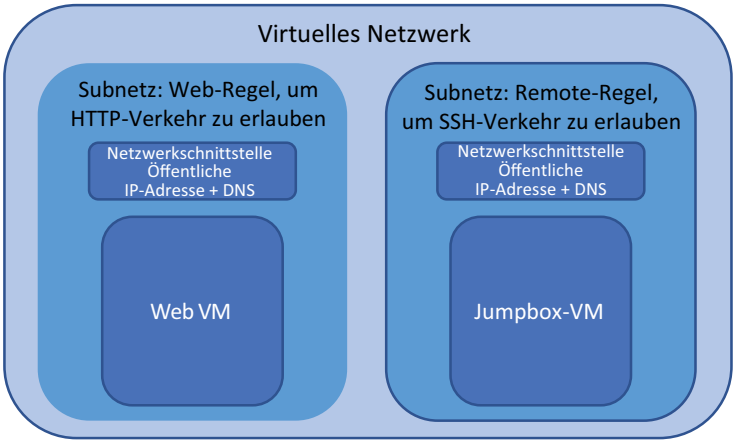


Abbildung 5.1 Softwaredefinierte Netzwerkverbindungen in Azure

Einige der Netzwerkkomponenten werden abstrahiert, wenn Sie PaaS-Ressourcen verwenden. Die Hauptkomponenten, die Sie für VMs verwenden, sind:

- Virtuelle Netzwerke und Subnetze (einschließlich IP-Adresspools)
- Virtuelle Netzwerkschnittstellenkarten
- Mindestens eine öffentliche IP-Adresse
- Interner DNS-Name und optionale öffentliche DNS-Namen für die externe Namensauflösung
- Netzwerksicherheitsgruppen und -regeln, die den Fluss des Netzwerkverkehrs wie eine normale Firewall sichern und steuern

5.1.1 Virtuelle Netzwerke und Subnetze

Beim Erstellen der VM in Kapitel 2 mussten Sie keine Netzwerkparameter anpassen. Die Azure-Plattform kann diese Ressourcen für Sie mit Standardnamen und IP-Adressumfang erstellen. In diesem Abschnitt erstellen wir die Netzwerkressourcen im Voraus und sehen uns an, wie sie für eine VM zusammenkommen.

Jetzt testen

Es ist oft einfacher, sich die Vernetzung vorzustellen, wenn Sie sie in Aktion erleben. Sie werden das Azure Portal für den Einstieg verwenden. Dies erfordert einige separate Schritte, die Leistungsfähigkeit der Azure-Befehlszeilenschnittstelle zeigt sich aber später in diesem Kapitel.

Machen Sie sich jetzt nicht zu viele Gedanken darüber, wie Sie Ihre eigenen Adressräume oder benutzerdefinierten DNS-Namen verwenden können. Um Ihr virtuelles Netzwerk und Subnetz einzurichten, führen Sie die folgenden Schritte aus:

1. Öffnen Sie das Azure Portal und wählen Sie oben links im Dashboard „Create a Resource“ (Eine Ressource erstellen) aus.
2. Wählen Sie „Networking“ (Netzwerke) aus der Liste der Marketplace-Dienste und dann „Virtual Network“ (Virtuelles Netzwerk).
3. Geben Sie einen Namen für das virtuelle Netzwerk ein, wie `vnetmol`.
4. Verschaffen Sie sich etwas mehr Spielraum, indem Sie den Adressraum in `10.0.0.0/16` ändern.

IP-Adressbereiche

Virtuelle Netzwerke decken eine bestimmte Bandbreite von IPs ab – einen Adressraum. Wenn Sie schon einmal eine IP-Adresse gesehen haben, ist Ihnen vielleicht die Subnetzmaske aufgefallen: oft etwas wie `255.255.255.0`. Diese Subnetzmaske wird oft in einer Kurzform verwendet, die angibt, wie groß dieser Bereich ist, z. B. `/24`.

Für das Azure Portal ist standardmäßig ein `/24`-Adressraum eingestellt. Sie möchten hier die Anzahl von zusätzlichen IP-Bereichen ohne viel Netzwerkwissen erhöhen, sodass Sie den Adressraum auf `/16` erhöhen. Sie geben diese Art von IP-Adresse nicht direkt an eine VM weiter – im nächsten Schritt erstellen Sie ein Subnetz, das einen kleineren Teil dieses Adressraums abdeckt.

Falls Ihnen Netzwerkadressenräume völlig fremd sind – keine Sorge. In den meisten Fällen werden Sie nicht täglich mit ihnen zu tun haben. Sensible Azure-Governance funktioniert ggf. wie in Ihrer bestehenden On-Premises-IT-Umgebung – eine Gruppe verwaltet die virtuellen Netzwerke von Azure, und Sie legen Ihre Anwendungen in einen bereits vorhandenen Raum ab.

5. Erstellen Sie eine Ressourcengruppe, z. B. `azuremolchapter5`, und wählen Sie dann eine Azure-Region in Ihrer Nähe aus.

6. Geben Sie einen Subnetznamen an, z. B. `websubnet`, und anschließend den Subnetz-Adressbereich `10.0.1.0/24`. Dieser Adressbereich ist Teil des zuvor angegebenen breiteren virtuellen Netzwerks. Später fügen Sie ein weiteres Subnetz hinzu.
7. Sehen Sie sich einige der anderen Optionen an, z. B. DDoS-Schutz (Distributed Denial of Service), Dienstendpunkte und Azure Firewall. Belassen Sie es vorerst bei den Standardeinstellungen. Ich hoffe allerdings, dieses Beispiel bietet Ihnen einige Hinweise darauf, was über ein einfaches virtuelles Netzwerk hinaus möglich ist.
8. Wenn Sie bereit sind, erstellen Sie das virtuelle Netzwerk und Subnetz.

5.1.2 Virtuelle Netzwerkschnittstellenkarten

Sie haben nun ein virtuelles Netzwerk und Subnetz erstellt und müssen eine VM verbinden. Genau wie bei einem herkömmlichen Desktop-PC, Laptop oder Tablet nutzen Sie eine Netzwerkschnittstellenkarte (NIC), um sich mit dem virtuellen Netzwerk zu verbinden. Und nein, es gibt kein kostenloses WLAN. Aber es gibt VM-Größen bei Azure, die derzeit bis zu acht NICs eine Geschwindigkeit von bis zu 32 Gbit/s ermöglichen. Selbst wenn ich nicht gut in Mathe wäre, könnte ich Ihnen sagen, dass sich diese Zahlen zu einer beträchtlichen Bandbreite summieren!

Sie fragen sich wahrscheinlich, weshalb Sie diese Ressourcen im Voraus erstellen sollten. Sie können das alles beim Erstellen einer VM erledigen. Das stimmt, aber machen wir einen Schritt rückwärts und denken über Netzwerkressourcen als langlebige Ressourcen nach.

Netzwerkressourcen existieren auch separat von VM-Ressourcen und können über den Lebenszyklus einer bestimmten VM hinaus fortbestehen. Dieses Konzept ermöglicht es Ihnen, feste Netzwerkressourcen zu erzeugen sowie eine VM zu erstellen, zu löschen und erneut zu erstellen, die die gleichen Netzwerkressourcen wie IP-Adressen und DNS-Namen beibehält. Denken Sie an eine Lab-VM oder eine Entwicklungs- und Testumgebung. Sie können schnell die exakt gleiche Umgebung reproduzieren, da sich nur die VM ändert.

Jetzt testen

Führen Sie die folgenden Schritte aus, um eine NIC zu erstellen:

1. Wählen Sie im Azure Portal oben links im Dashboard „Create a Resource“ (Eine Ressource erstellen).
2. Suchen Sie „Network Interface“ (Netzwerkschnittstelle), und wählen Sie es aus. Wählen Sie anschließend „Create“ (Erstellen) aus.
3. Geben Sie einen Namen für Ihre Netzwerkschnittstelle ein, z. B. `webvnic`, und wählen Sie dann das virtuelle Netzwerk und das Subnetz aus, das Sie in der vorherigen Übung erstellt haben.
4. Ich habe bereits über langlebige Ressourcen gesprochen. Jetzt können Sie sehen, wie sie funktionieren. Erstellen Sie eine statische IP-Adresszuweisung, die die Adresse `10.0.1.4` verwendet.

TIPP Warum .4? Wie sieht es mit den ersten drei Adressen im Adressraum aus? Azure behält sich die ersten drei IP-Adressen in jedem Bereich für eigene Verwaltungs- und Weiterleitungszwecke vor. Die erste Adresse, die Sie in jedem Bereich verwenden können, ist „.4“.

5 Erstellen Sie vorerst keine Netzwerksicherheitsgruppe. Wir kommen in wenigen Minuten darauf zurück. Wenn Sie zu den coolen Kids gehören und alles über IPv6 wissen, können Sie das Feld „Private IP Address (IPv6)“ (Private IP-Adresse (IPv6)) ankreuzen und einen Namen angeben. Bleiben Sie andernfalls bei IPv4.

6 Wählen Sie die vorhandene Ressourcengruppe aus der vorherigen Übung aus, und erstellen Sie dann die NIC in derselben Region wie das virtuelle Netzwerk.

7 Erstellen Sie die NIC, wenn Sie bereit sind.

Rollentrennung bei Azure

Sie müssen keine anderen Computing-Ressourcen innerhalb der gleichen Ressourcengruppe wie Ihr virtuelles Netzwerk erstellen. Denken Sie an das Konzept der Azure-Governance, das wir vorhin behandelt haben. Vielleicht haben Sie eine Gruppe von Netzwerk-Ingenieur*innen, die alle virtuellen Netzwerkressourcen in Azure verwalten. Wenn Sie Ressourcen für Ihre Anwendungen wie VMs erstellen und verwalten, tun Sie das in Ihren eigenen Ressourcengruppen.

Wir besprechen in späteren Kapiteln einige der Sicherheits- und Richtlinienmerkmale von Azure, mit denen Sie definieren können, wer auf bestimmte Ressourcen zugreifen und sie bearbeiten kann. Die Idee dahinter ist, dass, wenn Sie nicht viel über Netzwerkressourcen wissen (oder wissen möchten), können Sie sich mit dem verbinden, was Sie erhalten, und fertig. Das Gleiche gilt für andere Ingenieur*innen oder Entwickler*innen – sie können vielleicht Ihre Anwendungsressourcen sehen, sie aber nicht bearbeiten oder löschen.

Diese Art von Governance-Modell in Azure ist schön und gut, aber achten Sie darauf, nicht in die Falle der Arbeit in Silos zu tappen. In großen Unternehmen ist es eventuell unvermeidlich, dass Sie durch Abteilungsabgrenzungen eingeschränkt sind. Doch einer der großen Vorteile von Cloud Computing-Anbietern wie Azure ist die Verkürzung der Zeit bis zur Bereitstellung von Anwendungen, da es nicht notwendig ist, das Anschließen und Konfigurieren physischer Netzwerkressourcen abzuwarten. Planen Sie die Erstellung und Konfiguration der Azure-Netzwerkressourcen. Sie sollten dann in der Lage sein, Ihre Anwendungsressourcen nahtlos zu erstellen und zu verwalten.

5.1.3 Öffentliche IP-Adresse und DNS-Auflösung

Auf Ihre Ressourcen kann noch niemand zugreifen, da ihnen keine öffentlichen IP-Adressen oder DNS-Namen zugewiesen sind. Auch hier sollten wir dem Prinzip der langlebigen Ressourcen folgen und so eine öffentliche IP-Adresse und einen öffentlichen DNS-Namen erstellen, um diese dann Ihrer Netzwerkschnittstelle zuzuweisen.

Jetzt testen

Um eine öffentliche IP-Adresse und DNS-Eingabe für Ihre Netzwerkschnittstelle zu erstellen, führen Sie die folgenden Schritte aus:

1. Wählen Sie im Azure Portal oben links im Dashboard „Create a Resource“ (Eine Ressource erstellen).
2. Suchen Sie nach „Public IP Address“ (Öffentliche IP-Adresse), und wählen Sie diese aus. Wählen Sie dann „Create“ (Erstellen).
3. Erstellen Sie eine einfache SKU und IPv4-Adresse. Standard-SKUs und IPv6-Adressen sind für den Einsatz mit Load Balancern bestimmt (Kapitel 8). Machen Sie sich jetzt keine allzu großen Gedanken um die Unterschiede.
4. Geben Sie einen Namen ein, z. B. webpublicip, der eine dynamische Zuweisung verwendet.

Zuweisungsarten für IP-Adressen

Durch eine dynamische Zuweisung wird eine öffentliche IP-Adresse zugewiesen, wenn die VM gestartet wird. Beim Anhalten der VM wird die öffentliche IP-Adresse wieder freigegeben. Hier gibt es einige wichtige Punkte:

- Sie haben keine öffentliche IP-Adresse, bis Sie sie einer VM zuordnen und diese starten.
- Die öffentliche IP-Adresse kann sich ändern, wenn Sie die VM anhalten, freigeben oder starten.

Eine *statische* Zuweisung bedeutet, dass Sie eine öffentliche IP-Adresse haben, die ohne eine zugehörige VM zugewiesen wird, und diese Adresse ändert sich nicht: Diese Zuweisung ist nützlich, wenn Sie ein SSL-Zertifikat verwenden, das einer IP-Adresse zugeordnet ist, oder einen benutzerdefinierten DNS-Namen und Datensatz, die auf die IP-Adresse verweisen.

Im Moment verwenden Sie eine einzelne VM. Für den Produktionseinsatz führen Sie Ihre Anwendung idealerweise auf mehreren VMs aus, denen ein Lastausgleich vorangestellt ist. In diesem Szenario wird die öffentliche IP-Adresse dem Lastausgleich zugeordnet und erzeugt zu diesem Zeitpunkt typischerweise eine statische Zuweisung.

5. Geben Sie einen eindeutigen DNS-Namen ein. Dieser Name bildet den vollständig qualifizierten Domänennamen (FQDN für Englisch „Fully Qualified Domain Name“) für Ihre Ressource, der auf der Azure-Region basiert, in der Sie sie erstellen. Wenn Sie einen DNS-Namen „azuremol“ beispielsweise im Osten der USA erstellen, wird der FQDN „azuremol.eastus. cloudapp.azure.com“.

DNS-Einträge

Wie sieht es mit einem benutzerdefinierten DNS-Namen aus? Der Standard-FQDN ist nicht gerade benutzungsfreundlich. Verwenden Sie eine statische öffentliche IP-Adresse und erstellen Sie dann einen CNAME-Datensatz in Ihrer registrierten DNS-Zone. Sie behalten die Kontrolle über den DNS-Datensatz und können so viele Einträge erstellen, wie Sie für Ihre Anwendungen möchten.

Als Beispiel könnten Sie in der DNS-Zone „manning.com“ einen CNAME für „azuremol“ erstellen, der auf eine statische öffentliche IP-Adresse in Azure verweist. Benutzende würden dann auf „azuremol.manning.com“ zugreifen, um zu Ihrer Anwendung zu gelangen. Diese Adresse ist viel benutzungsfreundlicher als webmol.eastus.cloudapp.azure.com!

6. Wählen Sie die vorhandene Ressourcengruppe aus der vorherigen Übung aus, und erstellen Sie dann die öffentliche IP-Adresse in derselben Region wie das virtuelle Netzwerk.
7. Wenn Sie bereit sind, erstellen Sie eine öffentliche IP-Adresse.
8. Verbinden Sie die öffentliche IP-Adresse und die DNS-Namensbezeichnung mit der Netzwerkschnittstelle, die Sie in Abschnitt 5.1.2 erstellt haben. Suchen Sie in der Navigationsleiste links im Azure Portal nach „Resource Group" (Ressourcengruppe), und wählen Sie diese Option aus. Wählen Sie dann die Ressourcengruppe aus, in der Sie Ihre Netzwerkressourcen erstellt haben, beispielsweise „azuremolchapter5".
9. Wählen Sie Ihre öffentliche IP-Adresse aus der Liste der Ressourcen aus, und wählen Sie dann „Associate" (Zuordnen) aus.
10. Wählen Sie die Verknüpfung mit einer Netzwerkschnittstelle (beachten Sie aber, mit welchen anderen Adressen Sie die öffentliche IP-Adresse verknüpfen können). Wählen Sie dann die von Ihnen erstellte Netzwerkschnittstelle, z. B. webvnic.

Nach wenigen Sekunden wird das Fenster mit der öffentlichen IP-Adresse aktualisiert und zeigt, dass die IP-Adresse nun mit Ihrer Netzwerkschnittstelle verbunden ist. Wenn Sie „Dynamic" (Dynamisch) als Zuweisungstyp gewählt haben, ist die IP-Adresse noch leer (siehe Abbildung 5.2). Denken Sie daran, dass eine öffentliche IP-Adresse zugewiesen wird, wenn eine zugehörige VM eingeschaltet wird.

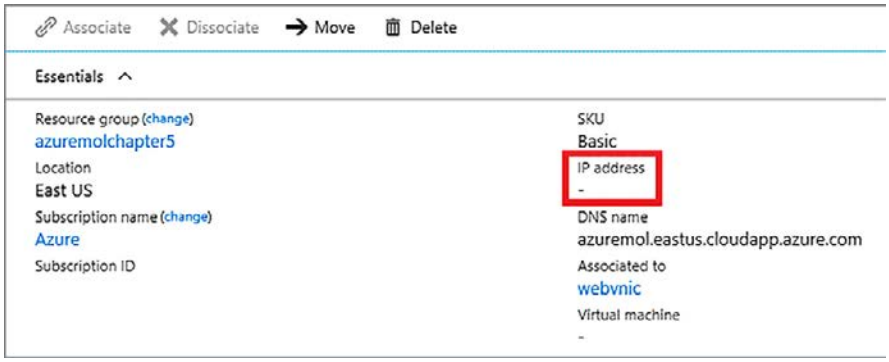


Abbildung 5.2 Die öffentliche IP-Adresse ist nun mit einer Netzwerkschnittstelle verbunden. Bei dynamischer Zuweisung wird keine öffentliche IP-Adresse angezeigt, bis eine VM erstellt und aktiviert wird.

5.2 Sicherung und Steuerung von Datenverkehr mit Netzwerksicherheitsgruppen

Preisfrage: Sollten Sie eine VM ohne Firewall mit dem Internet verbinden, um den Datenverkehrsfluss zu steuern und zu begrenzen? Wenn Sie mit „Sicher, warum nicht?" geantwortet haben, nehmen Sie sich bitte den Rest Ihrer Mittagspause Zeit, um ein wenig über Netzwerksicherheit im Wild Wide Web zu lesen.

Ich hoffe, Ihre Antwort war etwas wie: „*Nein!*“ Leider besteht eine zu große Gefahr für Ihre VM, kurz nach dem Einschalten Opfer eines automatisierten Cyberangriffs zu werden. Sie sollten immer die Best Practices befolgen, um das Betriebssystem und die Anwendungssoftware auf dem neuesten Stand zu halten, doch tatsächlich soll nicht einmal der Netzwerkverkehr Ihre VM erreichen, wenn das nicht notwendig ist. Ein normaler macOS- oder Windows-Computer verfügt über eine integrierte Softwarefirewall und jedes (kompetente) On-Premises-Netzwerk, das ich gesehen habe, hat eine Netzwerkfirewall zwischen dem Internet und dem internen Netzwerk. Bei Azure werden Firewall- und Traffic-Regeln durch Netzwerksicherheitsgruppen bereitgestellt.

5.2.1 Erstellen einer Netzwerksicherheitsgruppe

In Azure wendet eine NSG logisch eine Reihe von Regeln auf Netzwerkressourcen an. Durch diese Regeln wird bestimmt, welcher Datenverkehr in Ihre und aus Ihrer VM fließen kann. Sie legen fest, welche Ports, Protokolle und IP-Adressen erlaubt sind und in welche Richtung sie zugelassen sind. Diese Regelgruppen können auf eine einzelne Netzwerkschnittstelle oder ein ganzes Netzwerk-Subnetz angewendet werden. Durch diese Flexibilität können Sie die Art und den Zeitpunkt der Anwendung der Regeln für die Sicherheitsbedürfnisse Ihrer Anwendung genau kontrollieren.

Abbildung 5.3 zeigt den logischen Fluss eines eingehenden Netzwerkpakets beim Passieren einer NSG. Der gleiche Prozess würde für ausgehende Pakete gelten. Der Azure-Host unterscheidet nicht zwischen Datenverkehr aus dem Internet und Datenverkehr von einem anderen Punkt in Ihrer Azure-Umgebung, zum Beispiel von einem anderen Subnetz oder einem virtuellen Netzwerk. Für jedes eingehende Netzwerkpaket gelten die eingehenden NSG-Regeln und für jedes ausgehende die ausgehenden NSG-Regeln.

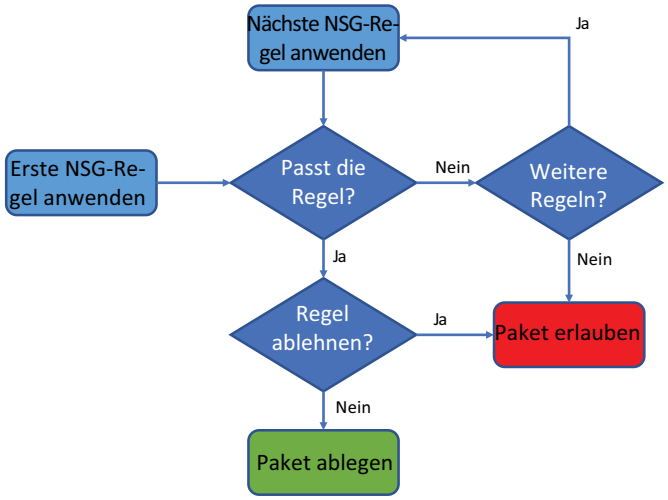


Abbildung 5.3 Eingehende Pakete werden geprüft und jede NSG-Regel wird entsprechend der Priorität angewendet. Bei einer Regelübereinstimmung mit „Allow“ oder „Deny“ („Zulassen“ oder „Blockieren“) wird das Paket entweder an die VM weitergeleitet oder gelöscht.

Folgendes passiert mit jedem Netzwerkpaket:

1. Die erste NSG-Regel wird angewendet.
2. Wenn die Regel nicht mit dem Paket übereinstimmt, wird die nächste Regel geladen, bis keine Regeln mehr vorhanden sind. Anschließend wird die Standardregel zum Löschen des Pakets angewendet.
3. Wenn eine Regel übereinstimmt, prüfen Sie, ob das Paket durch die Aktion abgelehnt werden soll. Wenn ja, wird das Paket gelöscht.
4. Andernfalls wird das Paket an die VM übergeben, wenn das Paket durch die Regel zugelassen werden soll.

Als Nächstes erstellen Sie eine NSG, damit diese Konzepte Sinn machen.

Jetzt testen

Führen Sie die folgenden Schritte durch, um eine Netzwerksicherheitsgruppe zu erstellen.

1. Wählen Sie im Azure Portal oben links im Dashboard „Create a Resource“ (Eine Ressource erstellen).
2. Suchen Sie nach und wählen Sie „Network Security Group“ (Netzwerksicherheitsgruppe) aus und dann „Create“ (Erstellen).
3. Geben Sie einen Namen ein, z. B. `webnsg`, und verwenden Sie die vorhandene Ressourcengruppe.

So einfach ist das. Der Großteil der Konfiguration für ein NSG folgt, wenn Sie die Filterregeln erstellen. In Abschnitt 5.2.2 besprechen wir, wie Sie das machen, und nehmen Ihre NSG in Betrieb.

5.2.2 Zuweisen einer Netzwerksicherheitsgruppe zu einem Subnetz

Die NSG trägt ohne Regeln nicht viel zum Schutz Ihrer VMs bei. Sie müssen sie auch mit einem Subnetz verknüpfen, so wie Sie Ihre öffentliche IP-Adresse zuvor mit einer Netzwerkschnittstelle verknüpft haben. Ordnen Sie Ihre NSG zunächst einem Subnetz zu.

Jetzt testen

Um Ihr virtuelles Netzwerk-Subnetz mit Ihrer Netzwerksicherheitsgruppe zu verknüpfen, führen Sie die folgenden Schritte aus.

1. Suchen Sie in der Navigationsleiste links im Azure Portal nach „Resource Group” (Ressourcengruppe), und wählen Sie diese Option aus. Wählen Sie dann die Ressourcengruppe aus, in der Sie Ihre Netzwerkressourcen erstellt haben, beispielsweise „azuremolchapter5“.
2. Wählen Sie Ihre NSG, z. B. „webnsg“.
3. Links unter „Settings“ (Einstellungen) finden Sie die Optionen „Network Interfaces“ (Netzwerkschnittstellen) und „Subnets“ (Subnetze). Wählen Sie „Subnets“ (Subnetze).
4. Wählen Sie die Schaltfläche „Associate“ (Zuordnen) und dann Ihr virtuelles Netzwerk und Netzwerk-Subnetz aus, das Sie zuvor erstellt haben. Wählen Sie dann „OK“, um Ihre NSG mit dem Subnetz zu verbinden.

Die Flexibilität der NSGs bedeutet, dass Sie mehrere Subnetze über verschiedene virtuelle Netzwerke hinweg mit einer einzelnen NSG verbinden können. Es handelt sich um eine Zuordnung von einem zu vielen, mit der Sie Kernregeln für die Netzwerksicherheit definieren können, die für eine Vielzahl von Ressourcen und Anwendungen gelten.

Nun sehen Sie, wie Ihre NSG aussieht und welche Standardregeln angewendet werden.

5 Links in Ihrer NSG wählen Sie „Inbound Security Rules“ (Sicherheitsregeln eingehend) aus. Es werden keine Sicherheitsregeln aufgelistet, zumindest keine, die Sie erstellt haben.

6 Wählen Sie „Default Rules“ (Standardregeln) aus, um zu sehen, was die Azure-Plattform für Sie erstellt (siehe Abbildung 5.4).

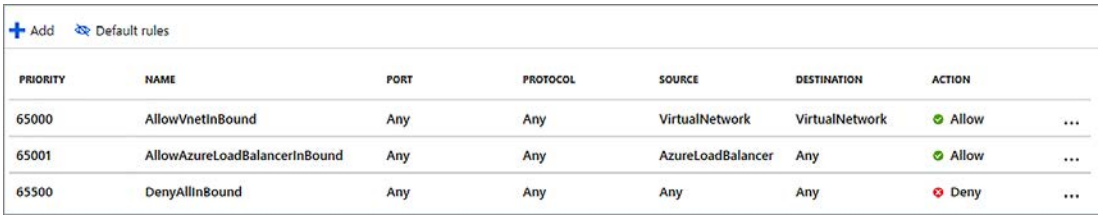


Abbildung 5.4 Es werden Standardsicherheitsregeln erstellt, die den internen Datenverkehr virtueller Netzwerke oder des Lastausgleichs zulassen, aber allen sonstigen Datenverkehr blockieren.

Drei Standardregeln wurden für Sie erstellt. Es ist wichtig, diese Regeln zu verstehen:

- *AllowVnetInBound* – erlaubt den gesamten Datenverkehr intern zum virtuellen Netzwerk. Wenn Sie mehrere Subnetze in Ihrem virtuellen Netzwerk haben, wird der Datenverkehr standardmäßig nicht gefiltert und wird zugelassen.
- *AllowAzureLoadBalancerInBound* – erlaubt den gesamten Datenverkehr von einem Azure-Lastausgleich zu Ihrer VM. Wenn Sie einen Lastausgleich zwischen Ihren VMs und dem Internet platzieren, stellt diese Regel sicher, dass der Datenverkehr aus dem Lastausgleich Ihre VMs erreichen kann, zum Beispiel bei der Überwachung eines Takts.
- *DenyAllInBound* – die letzte Regel, die angewendet wird. Löscht eingehende Pakete, die bis zu diesem Punkt kommen. Wenn es keine vorherigen Regeln zum Zulassen gibt, löscht diese Regel standardmäßig den gesamten Datenverkehr. Sie müssen also nur gewünschten bestimmten Datenverkehr zulassen.

Die Priorität einer NSG-Regel ist wichtig. Wenn eine „Allow“- oder „Deny“-Regel („Zulassen“ oder „Blockieren“) gilt, werden keine weiteren Regeln angewendet. Die Regeln werden in aufsteigender numerischer Prioritätsreihenfolge angewendet. Eine Regel mit einer Priorität von 100 wird beispielsweise vor einer mit 200 angewendet.

Wie schon bei früheren Diskussionen über die Governance von Azure-Ressourcen können diese NSG-Regeln bereits für Sie erstellt und auf ein bestimmtes Subnetz angewendet worden sein. Sie erstellen Ihre VMs und führen Ihre Anwendungen aus, und jemand anderes verwaltet die NSGs.

Es ist wichtig zu verstehen, wie der Datenverkehr fließt, falls etwas schief geht. Einige Tools in Azure können Ihnen dabei helfen, zu bestimmen, warum der Datenverkehr Ihre Anwendung nicht erreichen kann, wenn Sie denken, dass er das sollte.

5.2.3 Erstellen von Filterregeln für Netzwerksicherheitsgruppen

Sie haben Ihre NSG mit dem Netzwerk-Subnetz verknüpft und die Standardregeln überprüft. Erstellen wir nun eine grundlegende NSG-Regel, die HTTP-Verkehr ermöglicht.

Jetzt testen

Führen Sie die folgenden Schritte durch, um Ihre eigenen Regeln mit der Netzwerksicherheitsgruppe zu erstellen.

1. Um eine NSG-Regel aus dem vorherigen Fenster des Azure Portals zu erstellen, wählen Sie im Abschnitt „Inbound Security Rules“ (Sicherheitsregeln eingehend) die Option „Add“ (Hinzufügen).
2. Sie haben zwei Möglichkeiten, Regeln zu erstellen: Basic und Advanced (Erweitert). Um schnell vorgefertigte Regeln zu erstellen, wählen Sie „Basic“ oben im Fenster.
3. Wählen Sie HTTP aus dem Dropdown-Menü „Service“ (Dienst). Es werden viele Standarddienste angeboten, z. B. SSH, RDP und MySQL. Wenn Sie einen Dienst auswählen, wird der entsprechende Portbereich angewendet – in diesem Fall Port 80. Die standardmäßige Maßnahme für grundlegende Regeln ermöglicht den Datenverkehr.
4. Jeder Regel wird ein Prioritätswert zugeordnet. Je niedriger die Zahl, desto höher die Priorität. Behalten Sie die standardmäßige niedrige Priorität wie 100 bei.
5. Akzeptieren Sie den Standardnamen oder geben Sie einen eigenen an. Wählen Sie dann „OK“.

5.3 Entwickeln einer Beispielwebanwendung mit sicherem Datenverkehr

Bisher haben Sie ein virtuelles Netzwerk und ein Subnetz erstellt. Dann haben Sie eine Netzwerkschnittstelle erstellt und eine öffentliche IP-Adresse sowie eine DNS-Namensbezeichnung zugeordnet. Sie haben eine NSG erstellt und auf das gesamte Subnetz angewendet. Außerdem haben Sie eine NSG-Regel erstellt, um HTTP-Verkehr zu ermöglichen. Jetzt fehlt noch eins: die VM.

5.3.1 Erstellen von Netzwerkverbindungen für den Remote-Zugriff

In der Produktion sollten Sie keinen Remote-Zugriff, z. B. durch SSH oder RDP, auf VMs zulassen, auf denen Ihre Anwendungen ausgeführt werden. Sie haben in der Regel eine separate Jumpbox-VM, mit der Sie sich über das Internet verbinden, und greifen dann über die interne Verbindung auf weitere VMs zu. Bisher haben Sie alle virtuellen Netzwerkressourcen im Azure Portal erstellt. Steigen wir nun auf die Azure CLI um und sehen uns an, wie schnell Sie diese Ressourcen über die Befehlszeile erstellen können.

Jetzt testen

Sie haben die erste NSG im Azure Portal erstellt. Führen Sie die folgenden Schritte aus, um mit der Azure CLI eine weitere NSG zu erstellen:

1. Wählen Sie oben im Dashboard des Azure Portals das Cloud Shell-Symbol. Vergewissern Sie sich, dass die Bash-Shell geöffnet wird, nicht PowerShell.
2. Erstellen Sie eine zusätzliche NSG in der vorhandenen Ressourcengruppe. Wie in den vorherigen Kapiteln sollen die Backslashes (\) in den folgenden Befehlsbeispielen bei Zeilenumbrüchen helfen – Sie müssen diese nicht eintippen, wenn Sie nicht möchten. Geben Sie einen Namen ein, z. B. „remotensg“:

```
az network nsg create \
  --resource-group azuremolchapter5 \
  --name remotensg
```

3. Erstellen Sie eine NSG-Regel in der neuen NSG zum *Zulassen von* Port *22*. Geben Sie die Ressourcengruppe und NSG an, die Sie im vorherigen Schritt erstellt haben, zusammen mit einem Namen wie „allowssh“:

```
az network nsg rule create \
  --resource-group azuremolchapter5 \
  --nsg-name remotensg \
  --name allowssh \
  --protocol tcp \
  --priority 100 \
  --destination-port-range 22 \
  --access allow
```

4. Erstellen Sie ein Netzwerk-Subnetz für Ihre Remote-VM. Geben Sie einen Subnetznamen an, z. B. „remotesubnet“, zusammen mit einem Adresspräfix im Bereich des virtuellen Netzwerks, z. B. `10.0.2.0/24`. Sie hängen außerdem die NSG, die Sie in Schritt 3 erstellt haben, an das Subnetz an, z. B. „remotensg“:

```
az network vnet subnet create \
  --resource-group azuremolchapter5 \
  --vnet-name vnetmol \
  --name remotesubnet \
  --address-prefix 10.0.2.0/24 \
  --network-security-group remotensg
```

Drei Befehle: Das ist alles, was Sie zur Erstellung eines Subnetzes, einer NSG und einer Regel benötigen. Wird Ihnen das Potenzial der Azure CLI schon bewusst? Die Azure PowerShell ist genauso leistungsstark, fühlen Sie sich also nicht dazu gezwungen, alle Ressourcen im Azure Portal zu erstellen. Im weiteren Verlauf des Buchs werden Sie in den meisten Fällen eher die Azure CLI als das Portal verwenden.

5.3.2 Erstellen von VMs

Mit all den eingesetzten Netzwerkkomponenten können Sie nun zwei VMs erstellen. Eine VM wird im Subnetz erstellt, das HTTP-Verkehr ermöglicht, sodass Sie einen Webserver installieren können. Die andere VM wird im Subnetz erstellt, das SSH erlaubt, sodass Sie eine Jumpbox haben, um Ihre Anwendungsumgebung weiter zu sichern und eine Produktionsbereitstellung zu replizieren. Abbildung 5.5 erinnert daran, was Sie entwickeln.

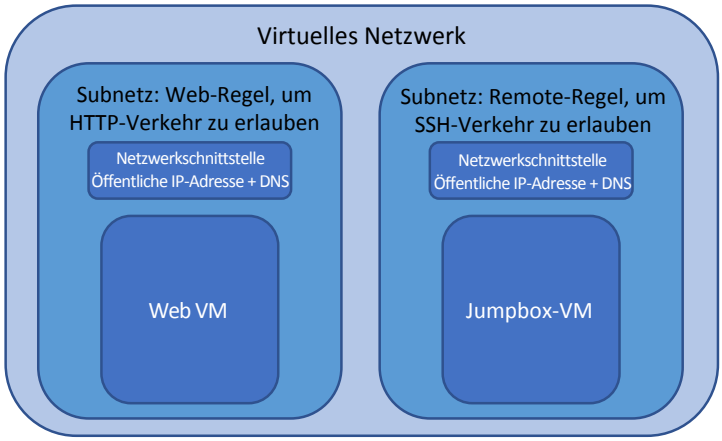


Abbildung 5.5 Sie vereinen zwei Subnetze, NSGs, Regeln, Netzwerkschnittstellen und VMs. Dieses Beispiel ähnelt einer produktionsfähigen Bereitstellung, bei der eine VM den Webserver ausführt und die für den öffentlichen Datenverkehr zugänglich ist, und eine andere VM in einem separaten Subnetz für Remote-Verbindungen zur restlichen Anwendungsumgebung verwendet wird.

Wenn Sie eine VM erstellen, können Sie die virtuelle Netzwerkschnittstelle bereitstellen, die Sie zuvor erstellt haben. Wenn Sie diese Netzwerkressource nicht angegeben haben, erstellt die Azure CLI ein virtuelles Netzwerk, ein Subnetz und eine NIC für Sie mit integrierten Standardeinstellungen. Um schnell eine VM zu erstellen, ist das toll. Doch Sie möchten dem Prinzip der langlebigen Netzwerkressourcen folgen, die vielleicht ein anderes Team verwaltet und in denen Sie Ihre VMs erstellen.

Jetzt testen

Um die Azure CLI zum Erstellen Ihres Webservers und Ihrer Ihrer Jumpbox-VMs zu verwenden, führen Sie die folgenden Schritte aus:

1. Erstellen Sie die erste VM für Ihren Webserver, und geben Sie einen Namen an, z. B. „webvm“. Fügen Sie die Netzwerkschnittstelle hinzu, z. B. „webvnic“, und geben Sie ein Image ein wie UbuntuLTS. Geben Sie einen Benutzungsnamen an, z. B. „azuremol“. Beim letzten Schritt, `--generate-ssh-keys`, werden der VM die SSH-Schlüssel hinzugefügt, die Sie in Kapitel 2 erstellt haben:

```
az vm create \
  --resource-group azuremolchapter5 \
  --name webvm \
  --nics webvnic \
  --image UbuntuLTS \
  --size Standard_B1ms \
  --admin-username azuremol \
  --generate-ssh-keys
```

2. Erstellen Sie die zweite VM für die Jumpbox. Dieses Beispiel zeigt, wie Sie ein bestehendes Subnetz sowie eine vorhandene NSG verwenden können und ermöglichen, dass die Azure CLI die Netzwerkschnittstelle erstellt und die entsprechenden Verbindungen herstellt. Sie erstellen eine öffentliche IP-Adresse, z. B. „remotepublicip“, als Teil dieses Befehls:

```
az vm create \
  --resource-group azuremolchapter5 \
  --name remotevm \
  --vnet-name vnetmol \
  --subnet remotesubnet \
  --nsg remotensg \
  --public-ip-address remotepublicip \
  --image UbuntuLTS \
  --size Standard_B1ms \
  --admin-username azuremol \
  --generate-ssh-keys
```

Die Ausgabe beider Befehle zeigt eine öffentliche IP-Adresse an. Notieren Sie sich diese IP-Adressen. Wenn Sie in der nächsten Übung versuchen, eine SSH-Verbindung mit Ihrem ersten VM für den Webserver herzustellen, schlägt dies fehl. Warum? Sie können eine SSH-Verbindung zur Remote-VM erstellen, da Sie eine NSG-Regel erstellt haben, die nur HTTP-Datenverkehr zur Web-VM erlaubt.

5.3.3 Verwenden des SSH-Agents für die Verbindung zu Ihren VMs

Ich muss Sie in ein Stück Magie mit SSH einführen, das es Ihnen ermöglicht, Ihre Jumpbox korrekt zu verwenden und sich über das virtuelle Netzwerk von Azure mit der Web-VM zu verbinden: den *SSH-Agent*. Dieser Agent gilt nur für Linux-VMs, wenn Sie also hauptsächlich mit Windows-VMs und Remote-Desktop-Protokoll-Verbindungen arbeiten, machen Sie sich keine Gedanken, wenn das SSH-Gerede neu für Sie ist. Sie können RDP-Verbindungen mit Windows-VMs über Ihre Jumpbox mit den lokalen Remote- oder mit den Domänenzugangsdaten herstellen, wenn Sie den Server entsprechend konfigurieren.

Ein SSH-Agent kann Ihre SSH-Schlüssel speichern und nach Bedarf weiterleiten. In Kapitel 2 haben Sie ein öffentliches SSH-Schlüsselpaar erstellt, und ich habe die öffentlichen und privaten Schlüssel erwähnt. Der private Schlüssel bleibt auf Ihrem Computer. Nur der öffentliche Schlüssel wird auf die Remote-VMs kopiert. Obwohl der öffentliche Schlüssel beiden von Ihnen erstellten VMs hinzugefügt wurde, können Sie nicht einfach per SSH auf Ihre Jumpbox und dann auf die Web-VM zugreifen. Warum? Diese Jumpbox verfügt nämlich über keine Kopie Ihres privaten Schlüssels. Wenn Sie versuchen, die SSH-Verbindung über die Jumpbox herzustellen, hat sie keinen privaten Schlüssel zur Abstimmung mit dem öffentlichen Schlüssel auf der Web-VM für eine Authentifizierung.

Der private Schlüssel dient zum Schutz, also sollten Sie es sich nicht zu einfach machen und den privaten Schlüssel in die Jumpbox kopieren. Alle anderen Benutzer*innen, die auf die Jumpbox zugreifen, könnten potenziell eine Kopie Ihres privaten Schlüssels abrufen und dann Ihre Identität annehmen, wenn der Schlüssel verwendet wird. Hier kommt der SSH-Agent ins Spiel.

Wenn Sie den SSH-Agent in Ihrer Cloud Shell-Sitzung ausführen, können Sie Ihre SSH-Schlüssel hinzufügen. Um Ihre SSH-Verbindung zur Jumpbox herzustellen, spezifizieren Sie die Verwendung dieses Agents an, um Ihre Sitzung zu tunneln. Mit dieser Technik können Sie Ihren privaten Schlüssel effektiv über die Jumpbox verwenden, ohne ihn jemals zu kopieren. Wenn Sie über SSH von der Jumpbox auf die Web-VM zugreifen, tunnelt der SSH-Agent Ihren privaten Schlüssel über die Jumpbox und ermöglicht Ihnen eine erfolgreiche Authentifizierung.

Jetzt testen

Um SSH mit Ihrer Jumpbox-VM zu verwenden, führen Sie die folgenden Schritte aus:

1. Starten Sie in Cloud Shell den SSH-Agent wie folgt:

```
eval $(ssh-agent)
```

2. Fügen Sie dem Agenten folgendermaßen den SSH-Schlüssel hinzu, den Sie in Kapitel 2 erstellt haben:

```
ssh-add
```

3. Greifen Sie per SSH auf Ihre Jumpbox-VM zu. Spezifizieren Sie die Verwendung des SSH-Agents mit dem Parameter `-A`. Geben Sie Ihre eigene öffentliche IP-Adresse ein, die in der Ausgabe beim Erstellen der Jumpbox-VM angezeigt wurde:

```
ssh -A azuremol@<publicIpAddress>
```

4. Das war das erste Mal, dass Sie eine SSH-Verbindung zur Jumpbox-VM hergestellt haben, also akzeptieren Sie die Eingabeaufforderung zur Verbindung mit den SSH-Schlüsseln.
5. Erinnern Sie sich daran, wie Sie eine statische private IP-Adresse für die Web-VM in Abschnitt 5.1.2 erstellt haben? Diese statische Adresse macht die Verbindung per SSH dazu viel einfacher. Stellen Sie eine SSH-Verbindung zur Web-VM wie folgt her:

```
ssh 10.0.1.4
```

6. Akzeptieren Sie die Aufforderung, um die SSH-Verbindung fortzusetzen. Der SSH-Agent tunnelte Ihren privaten SSH-Schlüssel durch die Jumpbox und ermöglicht es Ihnen, sich erfolgreich mit der Web-VM zu verbinden. Was nun? Nun haben Sie ein Lab, um die Früchte dieser Arbeit auszukosten.

5.4 Übung: Installation und Testen des LAMP-Webservers

Sie haben die schwere Arbeit bereits im Laufe des Kapitels gemeistert. Mit dieser schnellen Übung können Sie die Installation eines Webservers vertiefen und die NSG-Regel auf Ihrer VM in Aktion erleben:

1. *Installieren Sie einen grundlegenden Linux-Webserver.* Erinnern Sie sich an Kapitel 2, in dem Sie eine SSH-Verbindung zum VM hergestellt und dann das LAMP-Webserverpaket mit `apt` installiert haben. Installieren und konfigurieren Sie den Standard-LAMP-Webstack über die SSH-Verbindung zu Ihrer Web-VM, die Sie in Abschnitt 5.3.2 erstellt haben.
2. *Navigieren Sie zur Standardwebsite.* Sobald der LAMP-Webstack installiert ist, öffnen Sie die DNS-Namensbezeichnung in Ihrem Webbrowser, die Sie eingegeben haben, als Sie eine öffentliche IP-Adresse in Abschnitt 5.1.3 erstellt haben. In meinem Beispiel war das„azuremol.eastus.cloudapp .azure.com“. Sie können auch die öffentliche IP-Adresse verwenden, die ausgegeben wurde, als Sie die Web-VM erstellt haben. Denken Sie aber daran: Die öffentliche IP-Adresse ist anders als die der Jumpbox-VM, zu der Sie eine SSH-Verbindung hergestellt haben.

Teil 2:

Hochverfügbarkeit und Skalierung

Jetzt kommen wir zum eigentlich interessanten Teil! Da Sie nun die wesentlichen Ressourcen in Azure kennen, können Sie sich Bereichen wie Redundanz, Lastausgleich und geografischer Verteilung von Anwendungen zuwenden. Hier wird es spannend, und die Themen, mit denen wir uns befassen werden, zeigen Ihnen Lösungen und Best Practices auf, die Sie in Ihren realen Bereitstellungen nutzen können. Azure bietet einige beeindruckende Funktionen, die Daten global replizieren, den Kundenverkehr auf die nächstgelegene Instanz Ihrer Anwendung verteilen und auf Basis der Nachfrage automatisch skalieren. Diese Funktionen bilden die Stärken des Cloud Computing und bringen wahren Wert für Ihre Arbeit.

6 Azure Resource Manager

Meist möchten Sie möglichst wenig Zeit damit verbringen, darüber nachzudenken, wie Sie eine Anwendungsumgebung bereitstellen, und stattdessen schnell mit der eigentlichen Bereitstellung beginnen. In vielen IT-Umgebungen werden vermehrt Teams für die Entwicklung und den Betrieb gebildet, die kooperieren und eng zusammenarbeiten. Das Schlagwort *DevOps* hat daher bei Konferenzen und in Blogs gerade Konjunktur.

Die DevOps-Kultur bietet nichts wirklich Neues oder Wegweisendes. Häufig war es jedoch so, dass verschiedene Teams nicht optimal zusammengearbeitet haben. Moderne Werkzeuge haben die DevOps-Bewegung gefördert, indem sie mit Lösungen für die kontinuierliche Integration und kontinuierliche Auslieferung (Continuous Integration/Continuous Delivery, CI/CD) die gesamte Bereitstellung von Anwendungsumgebungen auf Grundlage eines einzigen Code-Check-ins durch einen Entwickelnden automatisieren. In der Regel ist es das Betriebsteam, das die CI/CD-Pipelines erstellt und pflegt, mit denen Entwickler*innen Anwendungsupdates viel schneller testen und bereitstellen können.

Von zentraler Bedeutung für den Aufbau und Betrieb von Ressourcen ist das Bereitstellungsmodell von Azure Resource Manager, was Ihnen wahrscheinlich bislang nicht klar war. Resource Manager ist ein Ansatz für das Erstellen und Bereitstellen von Ressourcen sowie Automatisierungsprozessen und Vorlagen, die diese Bereitstellungen steuern. In diesem Kapitel lernen Sie, wie Sie Resource Manager-Funktionen nutzen, insbesondere Zugriffskontrollen und -sperren, konsistente Vorlagenbereitstellungen und automatisierte Multi-Tier-Rollouts.

6.1 Der Ansatz von Azure Resource Manager

Als Sie in den vorherigen Kapiteln eine VM oder Web App erstellt haben, haben Sie als Kernkonstrukt zunächst eine Ressourcengruppe erstellt, die all Ihre Ressourcen enthielt. Eine Ressourcengruppe ist für alle Ressourcen von zentraler Bedeutung: Eine VM, eine Web App, ein virtuelles Netzwerk oder eine Speichertabelle können außerhalb einer Ressourcengruppe nicht existieren. Die Ressourcengruppe ist aber mehr als nur ein Ort, an dem Sie Ihre Ressourcen organisieren – weitaus mehr. In diesem Abschnitt sehen wir uns an, was das zugrundeliegende Modell von Azure Resource Manager ist und warum es beim Entwickeln und Ausführen von Anwendungen wichtig ist.

6.1.1 Design für den Lebenszyklus der Anwendung

Sie entwickeln idealerweise keine Anwendung, die Sie anschließend nicht pflegen. In der Regel müssen Sie Updates entwickeln und bereitstellen, neue Pakete installieren, neue VMs hinzufügen und zusätzliche Web-App-Bereitstellungsslots erstellen. Möglicherweise müssen Sie die virtuellen Netzwerkeinstellungen und IP-Adressen ändern. Ich hatte in den vorherigen Kapiteln erwähnt, dass Ihre virtuellen Netzwerke auf Azure möglicherweise von einem anderen Team verwaltet werden. Sie müssen anfangen, darüber nachzudenken, wie Sie auf einer großen, globalen Ebene und im Hinblick auf alle Aspekte des Anwendungslebenszyklus und der Anwendungsverwaltung vorgehen.

Es gibt ein paar entscheidende Ansätze für die Gruppierung von Ressourcen bei Azure:

- *Alle Ressourcen für eine bestimmte Anwendung befinden sich in derselben Ressourcengruppe* – Abbildung 6.1 zeigt, dass dieser Ansatz für kleinere Anwendungen oder Entwicklungs- und Testumgebungen gut funktioniert. Wenn Sie keine großen Netzwerkbereiche freigeben müssen und den Speicher individuell verwalten können, können Sie alle Ressourcen an einem Ort erstellen und dann Updates und Konfigurationsänderungen in einem Arbeitsgang verwalten.

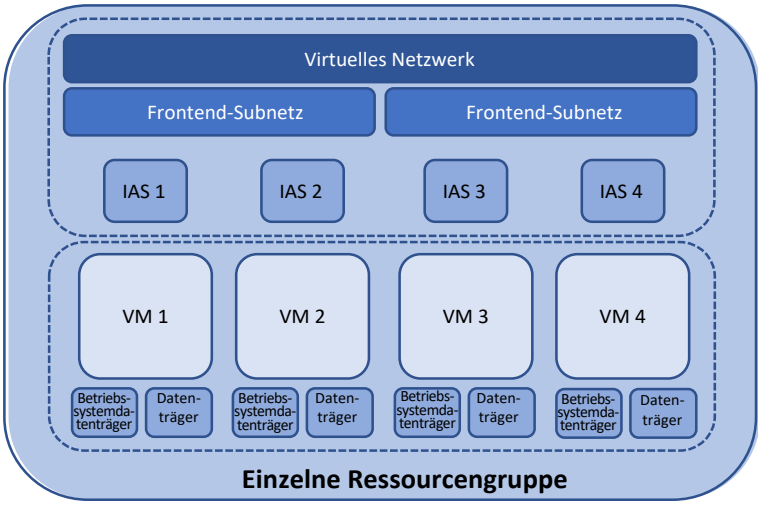


Abbildung 6.1 Eine Möglichkeit, eine Anwendung in Azure zu entwickeln, besteht darin, dass alle Ressourcen, die mit der Bereitstellung der Anwendung verbunden sind, in der gleichen

Ressourcengruppe erstellt und als eine Einheit verwaltet werden.

- *Vergleichbare Ressourcen, die nach Funktion in derselben Ressourcengruppe gruppiert sind* – wie Abbildung 6.2 zeigt, ist dieser Ansatz in größeren Anwendungen und Umgebungen häufiger anzutreffen. Ihre Anwendung kann in einer Ressourcengruppe existieren, die nur die VMs und unterstützende Anwendungskomponenten enthält. Virtuelle Netzwerkressourcen und IP-Adressen können sich in einer anderen Ressourcengruppe befinden, die von einer anderen Gruppe von Technikern gesichert und verwaltet wird.

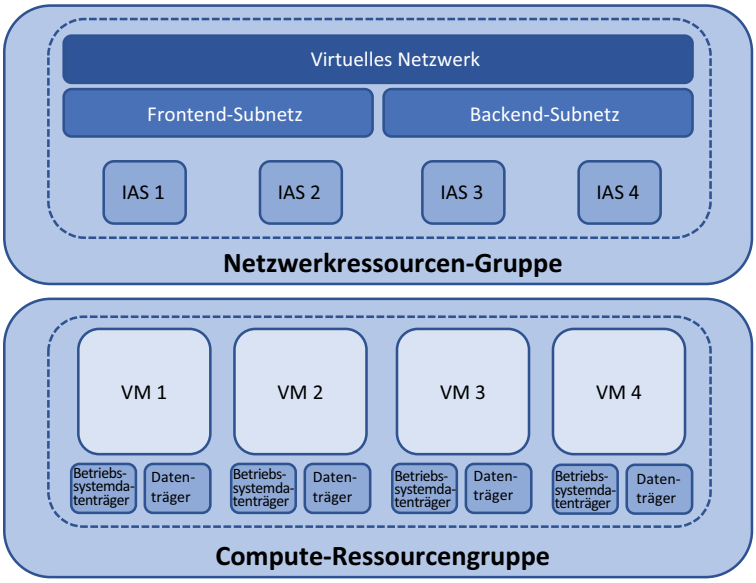


Abbildung 6.2 Ein alternativer Ansatz besteht darin, Ressourcen basierend auf ihrer Rolle zu erstellen und zu gruppieren. Gängig ist beispielweise, dass sich alle zentralen Netzwerkressourcen in einer anderen Ressourcengruppe befinden als die zentralen Rechenressourcen der Anwendung. Die VMs in der Ressourcengruppe für die Rechenressourcen können auf die Netzwerkressourcen in der separaten Gruppe zugreifen, die beiden Ressourcensätze können jedoch unabhängig verwaltet und gesichert werden.

Warum gibt es unterschiedliche Ansätze? Die Antwort dreht sich nicht um Arbeitsplatzsicherheit und die hübschen Silos, in denen einige Teams gerne arbeiten. Es geht vielmehr darum, wie Sie die zugrundeliegenden Ressourcen verwalten müssen. In kleineren Umgebungen und Anwendungen, in denen sich alle Ressourcen in der gleichen Ressourcengruppe befinden, sind Sie für alles in dieser Umgebung zuständig. Dieser Ansatz eignet sich auch gut für Entwicklungs- und Testumgebungen, in denen alles gebündelt ist. Alle Änderungen, die Sie am virtuellen Netzwerk vornehmen, wirken sich nur auf Ihre Anwendungs- und Ressourcengruppe aus.

Tatsächlich ändern sich Netzwerke nur selten. Die Adressbereiche sind meist klar definiert und geplant, sodass sie in Azure und an Bürostandorten auf der ganzen Welt gleichzeitig verwendet werden können. Logischerweise ist es oft sinnvoll, die Netzwerkkomponenten in einer eigenen Ressourcengruppe zu platzieren. Das

Netzwerk wird getrennt von der Anwendung verwaltet. Genauso kann der Speicher separat verwaltet und aktualisiert werden. An der Trennung solcher Ressourcen ist grundsätzlich nichts auszusetzen, solange die IT-Mitarbeiter*innen nicht zu einer Silo-Mentalität tendieren, die zu mangelnder Zusammenarbeit führt.

Für Ihre Anwendungen kann die Trennung von Ressourcen auch ein Vorteil sein, weil Sie Änderungen und Updates weitgehend unabhängig vornehmen können. Eben weil sich die Netzwerkkomponenten nicht in Ihrer Ressourcengruppe befinden, brauchen Sie sich keine Gedanken zu machen, wenn Sie Updates der Anwendung durchführen.

6.1.2 Schützen und Kontrollieren von Ressourcen

Auf jede Ressource kann eine andere Sicherheitsberechtigung angewendet werden. Diese Richtlinien legen fest, wer zu was berechtigt ist. Möchten Sie, dass jemand Ihre Web App neu startet oder die VM-Datenträger löscht? Und glauben Sie, dass Ihre netten Kolleg*innen im Netzwerkteam möchten, dass Sie ein neues virtuelles Subnetz des Netzwerks erstellen? Wahrscheinlich nicht. Bei Azure gibt es vier zentrale Rollen, die Sie Ressourcen zuweisen können. Diese ähneln Dateiberechtigungen:

- *Besitzer*in* – vollständige Kontrolle, im Grunde ein*e Administrator*in
- *Mitwirkende*r* – vollständige Verwaltung der Ressource, ausgenommen sind Änderungen der Sicherheits- und Rollenzuweisungen
- *Leser*in* – kann alle Informationen über die Ressource einsehen, aber keine Änderungen vornehmen
- *User Access-Administrator*in* – kann den Zugriff auf Ressourcen zuweisen oder entfernen.

Die rollenbasierte Zugriffssteuerung (RBAC) ist ein wesentliches Merkmal von Azure-Ressourcen und wird automatisch in die Benutzungskonten in Ihren Abonnements integriert. Denken Sie an die Dateiberechtigungen auf Ihrem normalen Computer. Die grundlegenden Dateiberechtigungen sind das Lesen, Schreiben und Ausführen. Wenn diese Berechtigungen kombiniert werden, können Sie für jede*n Benutzer*in oder jede Gruppe auf Ihrem Computer unterschiedliche Berechtigungsgruppen erstellen. Wenn Sie mit Netzwerkdateifreigaben arbeiten, dienen Berechtigungen häufig dazu, den Zugriff zu kontrollieren. Nach diesem Prinzip arbeitet auch RBAC in Azure, um den Zugriff auf Ressourcen zu kontrollieren – genau wie Dateiberechtigungen auf Ihrem lokalen Computer oder in der Netzwerkfreigabe (Abbildung 6.3).

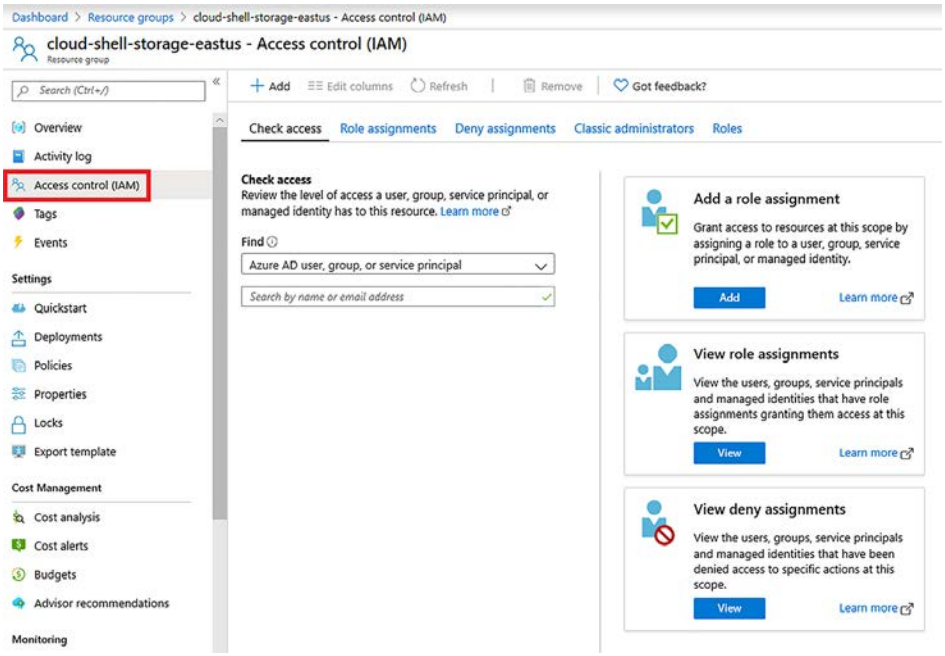


Abbildung 6.3 Die Zugriffssteuerung für die einzelnen Azure-Ressourcen listet die aktuellen Zuweisungen auf. Sie können Zuweisungen hinzufügen oder Rollen auswählen, um Informationen zu den verfügbaren Berechtigungssätzen anzuzeigen.

Jetzt testen

Öffnen Sie das Azure Portal in einem Webbrowser, und wählen Sie eine beliebige Ressource aus, die sich in Ihrem Besitz befindet, zum Beispiel den Cloud Shell-Speicher. Klicken Sie auf die Schaltfläche „Access Control (IAM)" (Zugriffssteuerung (IAM)), wie in Abbildung 6.3 dargestellt. Überprüfen Sie die aktuellen Rollenzuweisungen. Sehen Sie sich an, wie Sie eine Rollenzuweisung hinzufügen, und lernen Sie alle verfügbaren Rollenzuweisungen kennen. Über das Informationssymbol der einzelnen Rollen erfahren Sie, welche Berechtigungen der Rolle zugewiesen sind.

Wenn Sie die verfügbaren Rollen durchgehen, sehen Sie, dass es mehrere ressourcenspezifische Rollen gibt, einschließlich der Folgenden:

- Mitwirkende*r bei virtuellen Maschinen
- Websitemitwirkende*r
- Netzwerkmitwirkende*r

Erraten Sie, was diese Rollen bedeuten? Sie wenden die zentrale Rolle „Mitwirkender" der Plattform auf einen bestimmten Ressourcentyp an. Dieser Anwendungsfall geht auf das Konzept für die Verwaltung ähnlicher Ressourcen zurück. Ihnen könnten z. B. die Rollen „Mitwirkende*r für virtuelle Maschinen" und „Websitemitwirkende*r" zugewiesen werden. Alle in dieser Ressourcengruppe erstellten VMs oder Web Apps wären dann für Sie zur Verwaltung verfügbar. Sie könnten aber keine Netzwerkressourcen verwalten, die sich in einer anderen Ressourcengruppe befinden.

6.1.3 Schutz von Ressourcen durch Sperrung

Der berechtigungsbasierte Ansatz von RBAC eignet sich ausgezeichnet, um einzugrenzen, wer auf was zugreifen kann. Es können aber weiterhin Fehler passieren. Es hat einen guten Grund, warum Sie sich normalerweise nicht als Benutzer*in mit administrativen oder Root-Berechtigungen bei einem Server anmelden. Ein falscher Tastendruck oder Mausklick, und Sie haben versehentlich Ressourcen gelöscht. Selbst wenn Sie Sicherungen besitzen (Sie haben Sicherungen, oder? Und Sie testen diese regelmäßig?), ist dies ein zeitaufwändiger Prozess, der zu Produktivitäts- oder Umsatzeinbußen des Unternehmens führen kann. In Kapitel 13 werden Sie mehr darüber erfahren, wie die Azure Backup, Recovery und Replication Services Ihre Daten schützen.

Eine weitere Funktion des Resource Manager-Modells sind Ressourcensperren. Auf jede Ressource kann eine Sperre angewendet werden, die diese auf den reinen Lesezugriff beschränkt oder Löschoperationen verhindert. Die Löschsperre ist besonders hilfreich, weil es einfach extrem schnell passieren kann, dass die falsche Ressourcengruppe gelöscht wird. Sobald Sie einen Löschvorgang gestartet haben, gibt es kein Zurück mehr. Die Operation kann nicht mehr abgebrochen werden, nachdem die Azure-Plattform die Anforderung akzeptiert hat.

Für Produktionsworkloads rate ich Ihnen, für Ihre zentralen Ressourcen Sperren zu implementieren, um Löschungen zu verhindern. Diese gelten ausschließlich auf der Azure-Ressourcen- und Azure-Plattformebene, nicht für die Daten in Ihren Ressourcen. Sie können zum Beispiel Dateien auf einer VM oder eine Tabelle in einer Datenbank löschen. Die Azure-Ressourcensperren gelten nur, wenn Sie versuchen, die gesamte VM oder die Azure SQL-Datenbank zu löschen. Sie werden mir danken, wenn zum ersten Mal eine Sperre verhindert, dass Sie die falsche Ressourcengruppe löschen!

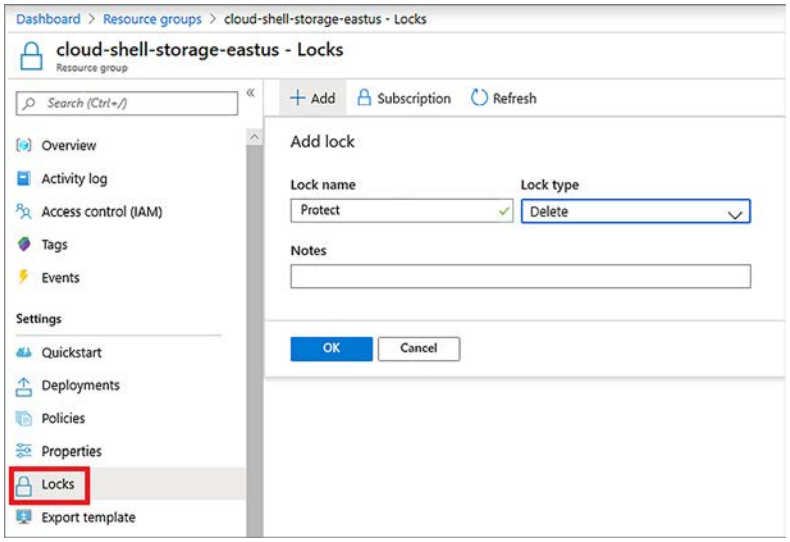


Abbildung 6.4 Erstellen einer Ressourcensperre im Azure Portal.

Jetzt testen

Führen Sie die folgenden Schritte aus, um die Azure-Ressourcensperren in Aktion zu erleben, wie in Abbildung 6.4 dargestellt:

1. Öffnen Sie das Azure Portal in einem Webbrowser, und wählen Sie eine beliebige Ressourcengruppe aus, die sich in Ihrem Besitz befindet, zum Beispiel den Cloud Shell-Speicher.
2. Wählen Sie auf der linken Seite des Portals „Locks" (Sperren) aus.
3. Geben Sie einen Sperrennamen ein, zum Beispiel „Protect", und wählen Sie im Dropdown-Menü „Lock Type" (Sperrentyp) aus. Wählen Sie dann „OK" aus. Ihre neue Sperre wird in der Liste angezeigt.
4. Wählen Sie die Übersicht für die Ressourcengruppe aus, und versuchen Sie dann, die Ressourcengruppe zu löschen. Sie müssen den Namen der Ressourcengruppe eingeben, um zu bestätigen, dass Sie sie löschen möchten (das ist eine gute Gedächtnisübung, die sicherstellt, dass Sie die richtige Ressource löschen!).
5. Klicken Sie auf die Schaltfläche „Delete" (Löschen), und sehen Sie sich die Fehlermeldung an, die anzeigt, dass Ihre Sperre Azure daran gehindert hat, die Ressource zu löschen.

6.1.4 Verwalten und Gruppieren von Ressourcen mit Tags

Eine letzte Funktion des Azure Resource Manager-Modells, das ich vorstellen möchte, sind *Tags*. An der Vorgehensweise zum Kennzeichnen von Ressourcen in Azure mit Tags ist nichts neu oder besonders, es handelt sich aber um ein häufig übersehenes Verwaltungskonzept. Sie können Tags auf eine Ressource in Azure anwenden, die deren Eigenschaften beschreiben, z. B. die Anwendung, zu der sie gehört, oder die dafür zuständige Abteilung oder ob es sich um eine Entwicklungs- oder Produktionsressource handelt.

Sie können Ressourcen anhand von Tags auswählen, um Sperren oder RBAC-Rollen anzuwenden oder Berichte über die Ressourcenkosten und den Verbrauch zu erstellen. Tags sind nicht spezifisch für eine Ressourcengruppe und können in Ihrem Abonnement wiederverwendet werden. Auf eine einzelne Ressource können bis zu 50 Tags angewendet werden. Dies bietet viel Flexibilität beim Markieren und Filtern markierter Ressourcen.

Jetzt testen

Führen Sie die folgenden Schritte aus, um Azure-Ressourcentags in Aktion zu erleben:

1. Öffnen Sie das Azure Portal in einem Webbrowser, und wählen Sie eine beliebige Ressource aus, zum Beispiel den Cloud Shell-Speicher. Obwohl Sie eine Ressourcengruppe selbst markieren können, wählen Sie für diese Übung keine Ressourcengruppe aus.
2. Wählen Sie mit der ausgewählten Ressource die Schaltfläche „Tags" aus, wie in Abbildung 6.5 dargestellt.
3. Geben Sie einen Namen, z. B. `Workload`, und einen Wert, z. B. `Entwicklung`, ein.
4. Wählen Sie „Save" (Speichern) aus.
5. Öffnen Sie Cloud Shell.
6. Benutzen Sie `az resource list` mit dem Parameter `--tag`, um Ressourcen anhand von Tags zu filtern. Geben Sie wie folgt Ihren eigenen Namen und Wert an:

```
az resource list --tag workload=development
```

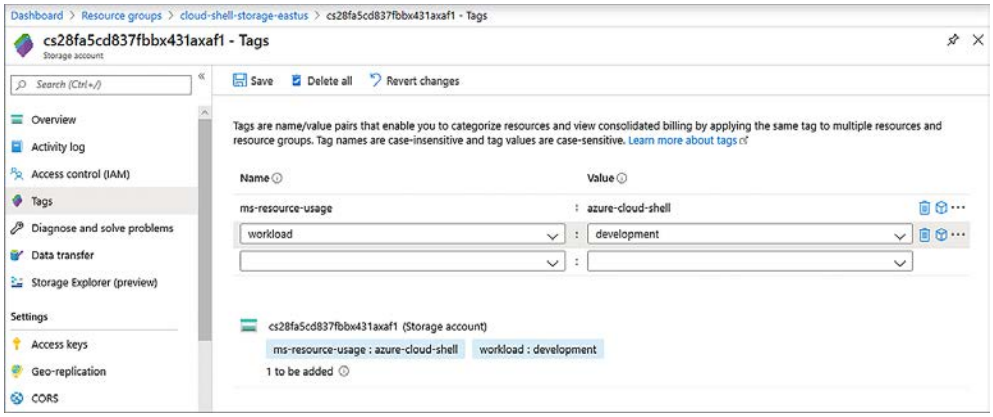


Abbildung 6.5 Sie können für jede Azure-Ressource oder Ressourcengruppe bis zu 50 aus Name und Wert bestehende Tags erstellen.

6.2 Azure Resource Manager-Vorlagen

Bis jetzt haben Sie eine kleine Anzahl von Azure-Ressourcen gleichzeitig erstellt. Hierzu haben Sie das Azure Portal oder die Azure CLI benutzt. Auch wenn ich Ihnen die Azure PowerShell nicht gezeigt habe, so habe ich in Kapitel 1 darüber gesprochen. Sie ist in der Cloud Shell verfügbar. Vielleicht haben Sie sie schon ohne mich ausprobiert. Das ist okay, ich fühle mich nicht ausgeschlossen! Wie ich in Kapitel 1 erwähnt hatte, bietet Azure Tools, mit denen Sie auswählen können, was für Sie und die Umgebung, in der Sie arbeiten, am bequemsten ist.

Die Kehrseite der Verwendung des Portals oder der CLI- oder PowerShell-Befehle ist, dass Sie im Webbrowser auf viele Schaltflächen klicken oder Befehlszeilen eingeben müssen, um Ihre Anwendungsumgebung zu erstellen. Sie könnten Skripte erstellen, die all diese Dinge ausführen, Sie müssten dann aber eine Logik entwickeln, die behandelt, wie mehrere Ressourcen gleichzeitig erstellt oder in welcher Reihenfolge die Ressourcen erstellt werden.

Ein Skript, das die Befehle der Azure CLI oder PowerShell aufnimmt, geht in die richtige Richtung, wenn es darum geht, wie Sie Anwendungsumgebungen entwickeln und bereitstellen sollten. Das gilt nicht nur für Azure, sondern für alle Plattformen. Es gibt eine Entwicklung hin zu Infrastructure as Code (IaC), was für die, die schon eine Weile in der IT arbeiten, nichts Neues ist. Es bedeutet nur, dass Sie sich nicht darauf verlassen, dass ein Mensch Befehle eintippt und eine Reihe von Schritten befolgt. Sie erstellen Ihre Infrastruktur stattdessen programmgesteuert aus einer Reihe von Anweisungen. Manuelle Bereitstellungen schließen die menschliche Komponente ein, was häufig zu kleineren Fehlkonfigurationen und Unterschieden in den endgültigen VMs führt, wie in Abbildung 6.6 dargestellt.

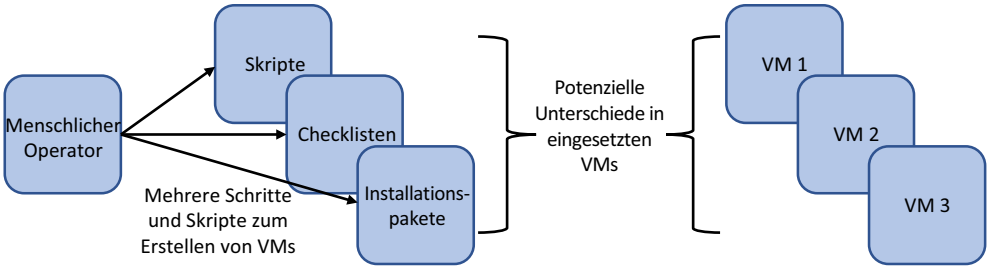


Abbildung 6.6 Menschen machen Fehler. Sie schreiben z. B. einen Befehl falsch oder überspringen einen Bereitstellungsschritt. Am Ende kann dies zu leicht unterschiedlichen VMs führen. Häufig wird Automatisierung genutzt, um den Menschen aus der Gleichung zu entfernen und so stets konsistente, identische Bereitstellungen zu erhalten.

Selbst wenn Sie über Skripts verfügen, brauchen Sie immer noch jemanden, der sie schreibt, pflegt und aktualisiert, sobald neue Versionen der Azure CLI- oder PowerShell-Module veröffentlicht werden. Ja, es gibt manchmal bahnbrechende Änderungen der Tools, um neue Funktionen einzuführen, auch wenn dies selten vorkommt.

6.2.1 Erstellen und Verwenden von Vorlagen

Resource Manager-Vorlagen können dazu beitragen, menschliche Fehler und die Abhängigkeit von manuell geschriebenen Skripts zu verringern. Vorlagen werden in JavaScript Object Notation (JSON) geschrieben, einem offenen Standard und plattformübergreifenden Format, das sich in einem einfachen Texteditor bearbeiten lässt. Mit Vorlagen können Sie konsistente, reproduzierbare Bereitstellungen erstellen, die Fehler minimieren. Durch eine weitere integrierte Vorlagenfunktion kann die Plattform Abhängigkeiten verstehen und Ressourcen, wenn möglich, parallel erstellen, um die Bereitstellungszeit zu verkürzen. Wenn Sie zum Beispiel drei VMs erstellen, müssen Sie nicht warten bis die Bereitstellung der ersten VM abgeschlossen ist, bevor Sie die zweite erstellen. Resource Manager kann alle drei VMs gleichzeitig erstellen.

Ein Beispiel für Abhängigkeiten ist das Erstellen einer virtuellen NIC, die Sie mit einem Subnetz verbinden müssen. Das Subnetz muss logischerweise vorhanden sein, bevor Sie die virtuelle NIC erstellen. Und das Subnetz muss Teil eines virtuellen Netzwerks sein, das daher vor dem Subnetz erstellt werden muss. Abbildung 6.7 zeigt die Verkettung der Abhängigkeiten. Wenn Sie versuchen, selbst ein Skript zu schreiben, müssen Sie die Reihenfolge, in der Ressourcen erstellt werden, sorgfältig planen und außerdem eine Logik integrieren, die in Erfahrung bringt, wann die übergeordneten Ressourcen bereit sind und Sie mit den abhängigen Ressourcen fortfahren können.

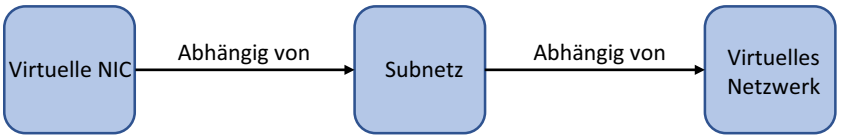


Abbildung 6.7 Azure Resource Manager übernimmt die Behandlung von Abhängigkeiten für Sie. Die Plattform weiß, in welcher Reihenfolge die Ressourcen erstellt werden müssen, und kennt den Status aller Ressourcen, ohne dass dafür die handgeschriebene Logik und Schleifen erforderlich sind, die Sie in Ihren eigenen Skripts brauchen.

Soll ich Ihnen etwas Interessantes sagen? Sie haben bereits Resource Manager-Vorlagen verwendet. In Kapitel 2 und bei der allerersten VM, die Sie erstellt haben. Wenn Sie im Portal oder in der Azure CLI eine VM erstellen, wird programmgesteuert eine Vorlage erstellt und bereitgestellt. Warum? Nun ja, warum sollte das Rad neu erfunden und die gesamte Logik für die Bereitstellung neu erstellt werden? Lassen Sie das Azure Resource Manager für Sie erledigen!

Hier sehen wir, wie eine Resource Manager-Vorlage aussieht. Die folgende Auflistung zeigt den Abschnitt, mit dem eine öffentliche IP-Adresse erstellt wird, so wie Sie es in früheren Beispielen getan haben, als Sie eine VM erstellt haben.

Auflistung 6.1 Erstellen einer öffentlichen IP-Adresse in einer Resource Manager-Vorlage

```
{
       "apiVersion": "2019-04-01",
       "type": "Microsoft.Network/publicIPAddresses",
       "name": "publicip",
       "location": "eastus",
       "properties": {
         "publicIPAllocationMethod": "dynamic",
         "dnsSettings": {
           "domainNameLabel": "azuremol"
    }
  }
},
```

Auch wenn Sie mit JSON nicht vertraut sind, das Format lässt sich halbwegs nachvollziehen. Sie definieren einen Ressourcentyp – in diesem Beispiel ist das `Microsoft.Network/publicIPAddresses`. Dann geben Sie einen Namen, hier `publicip`, und einen Standort an, hier `eastus`. Abschließend definieren Sie die Zuweisungsmethode, in diesem Beispiel ist das `dynamic`, und die DNS-Namensbezeichnung, die hier `azuremol` lautet. Das sind dieselben Parameter, die Sie im Azure Portal oder in der CLI angegeben haben. Was passiert, wenn Sie die PowerShell verwenden? Raten Sie mal. Sie werden nach denselben Parametern gefragt.

Der Unterschied zur Vorlage ist, dass Sie keine Informationen eingeben mussten. Die waren alle im Code enthalten. Vielleicht denken Sie jetzt „Toll, aber was ist, wenn ich jedes Mal verschiedene Namen benutzen will?“. Wie bei einem Skript können Sie diese mit Parametern und Variablen dynamisch zuweisen:

- *Parameter* sind Werte, die Sie eingeben müssen. Sie werden meist für Benutzeranmeldeinformationen, den VM-Namen und die DNS-Namensbezeichnung verwendet.
- *Variablen* kann vorab Werte zugewiesen werden. Sie werden aber ebenfalls bei jeder Verwendung der Vorlage angepasst, z. B. die VM-Größe oder der Name des virtuellen Netzwerks.

Jetzt testen

Wenn Sie sich eine komplette Resource Manager-Vorlage ansehen möchten, dann öffnen Sie unter http://mng.bz/QyWv das GitHub-Repository im Webbrowser.

6.2.2 Erstellen von Mehrlingen eines Ressourcentyps

Wenn Sie Ihre Vorlagen erstellen, müssen Sie vorausschauend darüber nachdenken, wie Ihre Anwendungen in Zukunft möglicherweise wachsen müssen. Wenn Sie Ihre Anwendung zum ersten Mal bereitstellen, benötigen Sie vielleicht nur eine einzelne VM. Wenn die Nachfrage nach der Anwendung aber wächst, müssen Sie zusätzliche Instanzen erstellen.

In einer herkömmlichen, auf Skripts basierenden Bereitstellung erstellen Sie eine `for`- oder `while`-Schleife, um mehrere Ressourcentypen zu erstellen. Im Resource Manager ist diese Funktionalität integriert! Genau wie in den meisten Programmier- und Skriptsprachen gibt es in Resource Manager über 50 Funktionstypen. Zu den gängigen Resource Manager-Funktionen gehören `length`, `equals`, `or` und `trim`. Die Anzahl der Instanzen, die erstellt werden, steuern Sie mit der Funktion `copy`.

Wenn Sie die Funktion `copy` verwenden, erstellt Resource Manager die von Ihnen angegebene Anzahl Ressourcen. Jedes Mal wenn Resource Manager die Operation zum Erstellen wiederholt, ist ein numerischer Wert verfügbar, mit dem die Ressourcen sequenziell benannt werden können. Auf diesen Wert können Sie mit der Funktion `copyIndex()` zugreifen. Das Beispiel in Auflistung 6.1 hat eine einzige öffentliche IP-Adresse erstellt. Das Beispiel in Auflistung 6.2 verwendet den gleichen `Microsoft.Network/publicIPAddresses`-Ressourcenanbietertyp, erstellt aber zwei öffentliche IP-Adressen. Mit copy definieren Sie, wie viele Adressen Sie erstellen möchten und mit `copyIndex()` geben Sie den Adressen sequentielle Namen.

Auflistung 6.2 Erstellen von mehreren öffentlichen IP-Adressen mit `copy`

```
{
       "apiVersion": "2019-04-01",
       "type": "Microsoft.Network/publicIPAddresses",
       "name": "[concat('publicip', copyIndex())]",
       "copy": {
         "count": 2
       }
       "location": "eastus",
       "properties": {
         "publicIPAllocationMethod": "dynamic",
       }
},
```

Sie benutzen außerdem die Funktion `concat`, um für jede erstellte Instanz den Namen der öffentlichen IP-Adresse mit dem numerischen Wert zu kombinieren. Nachdem diese Vorlage verwendet wurde, heißen Ihre beiden öffentlichen IP-Adressen `publicip0` und `publicip1`. Diese Namen sind nicht sehr beschreibend, aber dieses grundlegende Beispiel zeigt, wie Sie eine Konvention für die Nummerierung nutzen können, wenn Sie mit der Funktion `copy` mehrere Ressourcen erstellen.

6.2.3 Tools für das Erstellen eigener Vorlagen

Ich muss Ihnen etwas beichten: So elegant Resource Manager-Vorlagen sein mögen und auch wenn ich sie als beste Möglichkeit zur Bereitstellung von Anwendungen auf Azure empfehle – Sie müssen die Vorlagen auch weiterhin schreiben. Es gibt verschiedene Tools, die dies vereinfachen. Und es sind Hunderte von Mustervorlagen von Microsoft und Dritten verfügbar. Eine der besten Möglichkeiten zu lernen, wie Vorlagen erstellt und verwendet werden, besteht darin, die Schnellstartvorlagen zu analysieren, die Microsoft in seinen Beispielen unter https://github.com/Azure/azure-quickstart-templates zur Verfügung stellt.

Wenn Sie die Ärmel hochkrempeln und eigene Vorlagen schreiben wollen, empfehle ich Ihnen vor allem zwei Tools. Der erste ist Visual Studio Code, ein kostenloser Open-Source-Editor für mehrere Plattformen (https://code.visualstudio.com). Neben einigen integrierten Funktionen, z. B. die Quellcodeverwaltung und die GitHub-Integration, sind Erweiterungen verfügbar, die automatisch die verschiedenen Abschnitte oder Provider für die Ressourcen erstellen, aus denen eine Vorlage besteht, wie in Abbildung 6.8 dargestellt. Wenn Sie VS Code heruntergeladen und installiert haben, navigieren Sie zu „View" (Ansicht) > „Extensions" (Erweiterungen), und suchen Sie nach *Azure*.

Abbildung 6.8 In Visual Studio Code sind viele Erweiterungen verfügbar, die das Erstellen und Verwenden von Azure Resource Manager-Vorlagen erleichtern und optimieren.

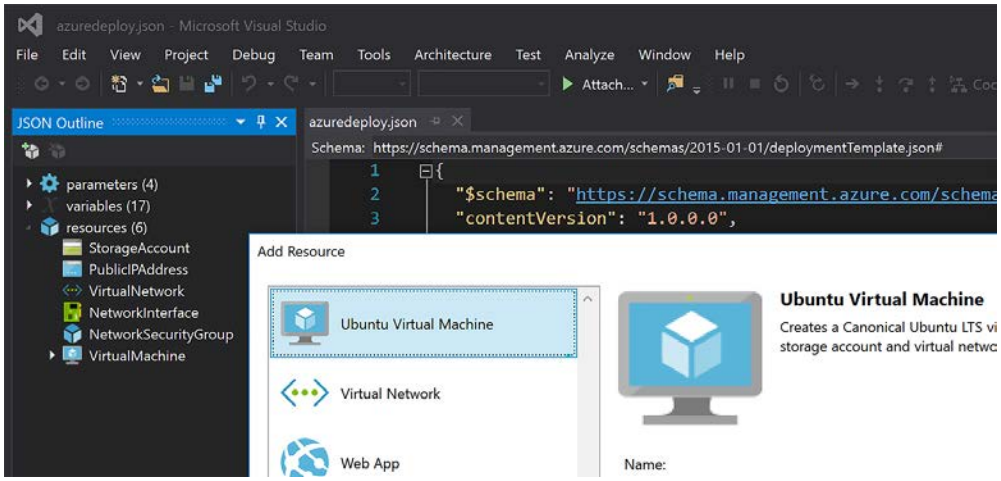


Abbildung 6.9 Mit Visual Studio können Sie Vorlagen grafisch erstellen und JSON-Ressourcen prüfen.

Eine stärker grafisch orientierte Möglichkeit, Azure Resource Manager-Vorlagen zu erstellen, bietet der Visual Studio-Editor, wie in Abbildung 6.9 dargestellt. Es gibt Versionen sowohl für Windows als auch für macOS, Sie benötigen für die Nutzung des Editors aber jeweils eine eigene Lizenz. Eine Community Edition ist verfügbar, passen Sie damit aber auf, wenn Sie Vorlagen in Ihrem Unternehmen erstellen. In der Regel benötigen Sie dann eine höher lizenzierte Version. Fragen Sie daher Ihre*n Lizenzexpert*in, da Visual Studio für Anwendungsentwickler*innen gedacht ist.

Sie können natürlich auch einen einfachen Texteditor verwenden. Azure Resource Manager-Vorlagen werden auch deshalb in JSON geschrieben, weil dazu keine speziellen Tools erforderlich sind. Die Arbeit mit JSON muss jedoch erst gelernt werden. Deshalb empfehle ich Ihnen, sich die Schnellstartvorlagen im Repository mit den Azure-Beispielen anzusehen. Achten Sie auf Einrückungen, Kommas am Ende und die Verwendung von Klammern, eckigen Klammern und geschweiften Klammern!

Leben auf dem Mars

Es gibt Tools von Drittanbietern und andere Möglichkeiten, Vorlagen in Azure zu verwenden. HashiCorp bietet viele Open Source-Tools und -Lösungen rund um Cloud Computing. Eine davon ist Terraform. Mit Terraform definieren Sie alle Ressourcen, die Sie erstellen möchten, auf die gleiche Weise, wie Sie eine native Azure Resource Manager-Vorlage erstellen. Sie können auch Abhängigkeiten definieren und Variablen verwenden. Der Unterschied ist, dass Terraform anbieterübergreifend arbeitet. So kann beispielsweise in Azure, Google Cloud, AWS und vSphere derselbe Ansatz für Konstruktionen und Vorlagen verwendet werden. Der Unterschied liegt in den Provisionern, die Sie für jede Ressource verwenden.

Ist das wirklich ein Ansatz, bei dem eine einzige Vorlage für jeden Anbieter funktioniert? Nein, ganz und gar nicht. Terraform ist auch eine Anwendung, die Ihre Vorlage analysiert und dann mit dem jeweiligen Cloud-Anbieter, z. B. Azure, kommuniziert. Sie erhalten keinerlei Bearbeitungsfunktionen, ganz zu schweigen von grafischen Werkzeugen, um Ihre Vorlage zu erstellen. Sie nehmen einen Editor und schreiben die Vorlage von Hand. Auch hier ist der beste Weg, um Terraform zu lernen, die zugehörige Dokumentation und die Beispielvorlagen zu erkunden.

Der Grund, warum ich dieses Thema anspreche, ist die Rückkehr zum besten Konzept bei Azure. Wenn Sie Azure Resource Manager-Vorlagen, die in JSON geschrieben wurden, etwas zu unhandlich finden, sehen Sie sich stattdessen ein Produkt wie Terraform an. Geben Sie die über Vorlagen gesteuerten Resource Manager-Bereitstellungen nicht auf. Vorlagen sind der beste Ansatz, um reproduzierbare, konsistente Bereitstellungen im großen Umfang durchzuführen. Finden Sie daher einen guten, vorlagengesteuerten Ansatz, der für Sie funktioniert.

6.2.4 Speichern und Verwenden von Vorlagen

Nun gefallen Ihnen die Azure Resource Manager-Vorlagen, und Sie haben Visual Studio oder Visual Studio Code installiert, um Ihre eigenen Vorlagen zu schreiben. Wie speichern Sie diese Vorlagen und stellen Sie sie dann bereit? In der praktischen Übung am Ende des Kapitels stellen Sie eine Vorlage aus dem Repository mit Azure-Beispielen auf GitHub bereit. Dies ist ein öffentliches Repository, und Sie möchten Ihre Anwendungsvorlagen vielleicht nicht der ganzen Welt zur Verfügung stellen.

Es gibt ein paar häufig angewendete Methoden für das private Speichern von Resource Manager-Vorlagen:

- Verwenden Sie ein privates Repository oder eine private Netzwerkdateifreigabe in Ihrem Unternehmen.
- Verwenden Sie Azure Storage, um Vorlagen für die Bereitstellung zentral zu speichern und zu schützen.

Es gibt kein richtiges oder falsches Verfahren für das Speichern und Bereitstellen von Vorlagen. Sie können flexibel alle bereits vorhandenen Prozesse und Tools nutzen. Der Vorteil der Verwendung eines Repositorys besteht darin, dass es in der Regel eine Art Versionskontrolle gibt, mit der Sie konsistente Bereitstellungen sicherstellen und die Historie Ihrer Vorlagen bei Bedarf prüfen können. Die einzige Einschränkung besteht darin, dass Sie bei der Bereitstellung der Vorlage die entsprechenden Anmeldeinformationen für den Zugriff auf den freigegebenen Standort angeben müssen. Dieser Authentifizierungsprozess kann unterschiedlich aussehen. Sie können zum Beispiel einen Benutzernamen oder ein Zugriffstoken als Teil der URL zu einer Vorlage in einem Repository angeben oder ein SAS-Token (Shared Access Signature) bereitstellen, wenn Sie Azure Storage verwenden.

Öffentliche Repositorys, wie GitHub, bieten auch eine gute Möglichkeit zu lernen und zu teilen. Ich rate Ihnen, Ihre Vorlagen für den Produktivbetrieb privat zu speichern. Wenn Sie aber zu Übungszwecken oder um neue Funktionen zu testen, eine gute Vorlage erstellt haben, dann teilen Sie sie auf GitHub, um der IT-Community etwas zurückzugeben und anderen zu helfen, die dieselben Bereitstellungen durchführen möchten. Und wenn Sie anfangen, eigene Vorlagen zu erstellen, dann prüfen Sie unbedingt, welche Vorlagen es bereits gibt, damit Sie nicht jedes Mal bei Null anfangen und das Rad neu erfinden!

6.3 Übung: Bereitstellen von Azure-Ressourcen mit einer Vorlage

All die Theorie rund um Bereitstellungsmodelle und -ansätze ist großartig, die Vorteile und die Effizienz werden Sie aber erst erleben, wenn Sie Vorlagen wirklich verwenden:

1. Gehen Sie zu den Azure Quickstart Templates auf GitHub (https://github.com/Azure/azure-quickstart-templates), und suchen Sie eine heraus, die Sie interessiert. Einen guten Ausgangspunkt bildet eine einfache Linux- oder Windows-VM.
2. In den GitHub-Beispielen befinden sich Schaltflächen für die direkte Bereitstellung auf Azure. Wenn Sie eine Vorlage gefunden haben, die Ihnen gefällt, klicken Sie auf „Deploy to Azure" (Bereitstellung in Azure), wie in Abbildung 6.10 dargestellt, und folgen Sie den im Portal angezeigten Schritten. Der Prozess stimmt in etwa mit dem der VM-Erstellung überein. Es sind aber nur wenige Eingabeaufforderungen für die erforderlichen Parameter erforderlich. Alle anderen Ressourcen werden für Sie erstellt und sind deshalb ausgeblendet.
3. Im letzten Schritt für die Bereitstellung Ihrer Vorlage müssen Sie die gezeigte Lizenzvereinbarung akzeptieren und dann „Purchase" (Kaufen) auswählen. Wenn Sie eine Vorlage bereitstellen, erstellen Sie Azure-Ressourcen. Mit „Purchase" (Kaufen) erklären Sie daher, dass Sie damit einverstanden sind, die Kosten für diese Azure-Ressourcen zu bezahlen.

   Eine der grundlegenden Vorlagen, z. B. eine einfache Linux- oder Windows-VM, kostet ungefähr so viel wie jede andere VM, die Sie bisher erstellt haben. Stellen Sie sicher, dass Sie die Ressourcengruppe nach Abschluss der Bereitstellung löschen. Genau so, wie Sie auch nach jeder anderen Übung aufräumen.

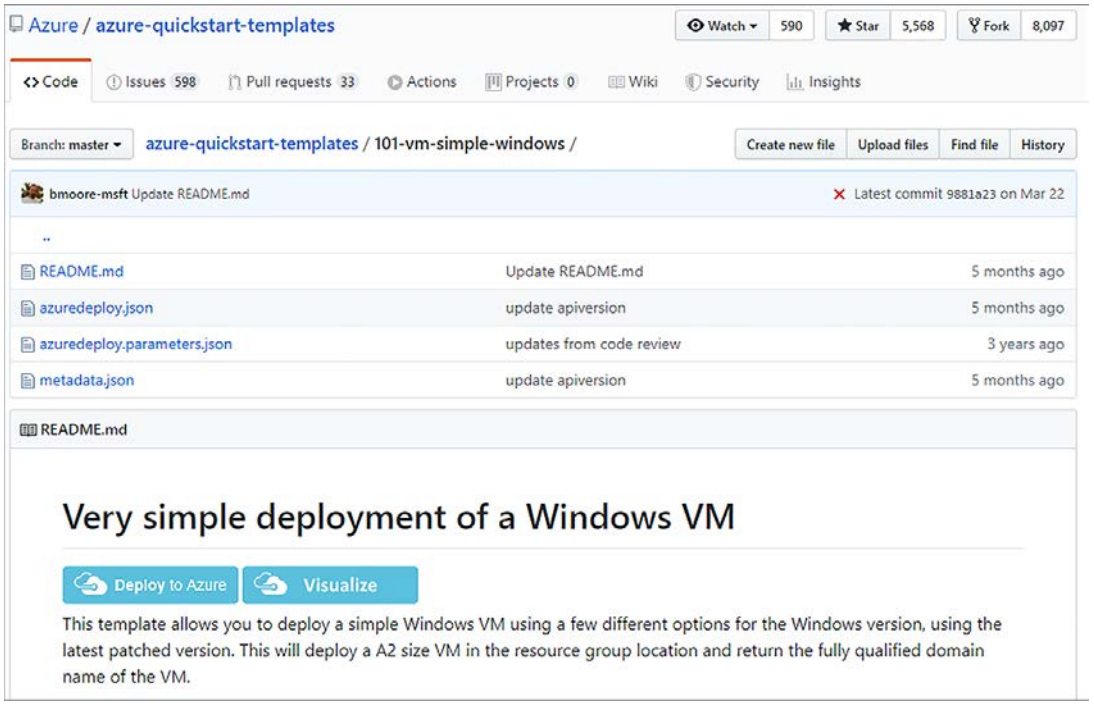


Abbildung 6.10 Jede im GitHub-Repository verfügbare Resource Manager-Vorlage besitzt eine Schaltfläche „Deploy to Azure“ (Bereitstellung in Azure). Wenn Sie auf diese Schaltfläche klicken, werden das Azure Portal und die Vorlage geladen. Sie werden aufgefordert, einige grundlegende Parameter einzugeben. Den Rest der Bereitstellung übernimmt dann die Vorlage.

Parameter in Vorlagen

Wie bereits in Abschnitt 6.2.1 angesprochen, können Sie in Ihren Vorlagen Parameter und Variablen verwenden. Denken Sie daran, dass Parameter Werte sind, zu deren Eingabe Sie aufgefordert werden, und dass Variablen dynamische Werte sind, die innerhalb einer Vorlage angewendet werden können. Die Werte, zu deren Eingabe Sie aufgefordert werden (Parameter), variieren von Vorlage zu Vorlage. Abhängig von der von Ihnen ausgewählten Schnellstartvorlage, werden Sie daher eventuell nach einem oder zwei Werten gefragt oder aber nach sieben oder acht.

Wenn Sie eigene Vorlagen entwerfen, versuchen Sie, vorherzusehen, wie Sie und andere Benutzer die Vorlage wiederverwenden möchten, wenn Sie Anwendungen bereitstellen. Sie können einen Standardwert angeben und die zulässigen Werte einschränken. Verfahren Sie umsichtig mit Standardwerten und zulässigen Werten. Andernfalls schränken Sie Benutzer*innen möglicherweise zu stark ein und zwingen diese dazu, eigene Vorlagen zu erstellen. Versuchen Sie möglichst, wiederverwendbare grundlegende Vorlagen zu erstellen, die genügend Flexibilität bieten.

4 Sobald Ihre Vorlage bereitgestellt wurde, kehren Sie zu GitHub zurück und prüfen Sie die Datei „azure-deploy.json”. Diese Datei ist die Azure Resource Manager-Vorlage, mit der das Beispiel erstellt und bereitgestellt wurde. Versuchen Sie, die verschiedenen Ressourcentypen und -konfigurationen zu verstehen, die angewendet wurden. Je öfter Sie mit Azure-Ressourcentypen und -Vorlagen arbeiten, desto besser verstehen Sie das JSON-Format. Ehrenwort!

7 Hochverfügbarkeit und Redundanz

Unzählige Male ist mir in der IT etwas kaputtgegangen. Einen Tag vor einer Konferenz ging die Festplatte meines Laptops kaputt, die Stromversorgung in einem E-Mail-Server qualmte und Netzwerkschnittstellen auf einem Core Router sind ausgefallen. Und über Betriebssystem-, Treiber- und Firmware-Updates spreche ich besser erst gar nicht! Ich bin mir sicher, dass jeder, der in der IT arbeitet, Horrorgeschichten über Situationen zum Besten geben kann, die er erlebt hat – in der Regel mit Problemen, die am späten Abend oder zu einem kritischen Zeitpunkt für das Unternehmen aufgetreten sind. Gibt es so etwas wie einen guten Ausfall, der zu einem schönen Zeitpunkt passiert?

Wenn Sie mit Ausfällen in der IT rechnen, lernen Sie, Ihre Anwendungen so zu planen und zu gestalten, dass sie sich an Probleme anpassen. In diesem Kapitel erfahren Sie, wie Sie Azure-Funktionen für Hochverfügbarkeit und Redundanz nutzen, um durch Wartungsupdates und -ausfälle verursachte Störungen zu minimieren. Dieses Kapitel bildet die Grundlage für die zwei oder drei nächsten Kapitel, in denen Sie anfangen werden, von einer Anwendung, die in einer einzigen VM oder Web App läuft, zu einer Anwendung überzugehen, die skaliert und global verteilt werden kann.

7.1 Die Notwendigkeit von Redundanz

Wenn Ihnen Kund*innen ihr wichtiges Pizza-Geschäft anvertrauen sollen, müssen Sie Anwendungen bereitstellen, die jederzeit erreichbar sind. Die meisten Kund*innen werden nicht auf einer Website nach den Betriebszeiten suchen – vor allem, wenn Sie in einem globalen Umfeld arbeiten und Ihre Kund*innen aus der ganzen Welt kommen. Wenn Kund*innen Hunger haben, dann wollen sie essen!

Abbildung 7.1 zeigt ein einfaches Beispiel für eine Anwendung, die auf einem einzigen virtuellen Computer läuft. Leider bildet diese Anwendung einen Single Point of Failure, also eine einzelne Fehlerquelle. Wenn diese eine VM nicht verfügbar ist, ist die Anwendung nicht verfügbar, was zu unzufriedenen und hungrigen Kund*innen führt.

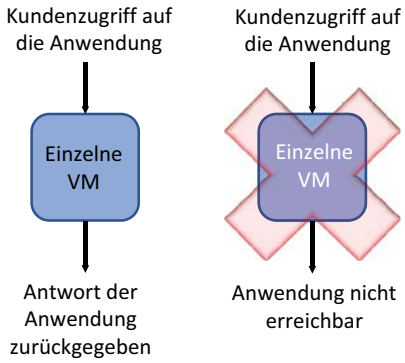


Abbildung 7.1 Wenn Ihre Anwendung auf einer einzigen VM läuft, führt jeder Ausfall dieser VM dazu, dass die Anwendung nicht mehr erreichbar ist. Das kann bedeuten, dass Kund*innen ihr Geschäft woanders betreiben oder zumindest mit dem von Ihnen gebotenen Service unzufrieden sind.

Für Ihr Auto haben Sie wahrscheinlich einen Ersatzreifen, falls Sie mal einen Platten haben sollten. Wenn Sie einen Laptop oder ein Tablet verwenden, werden Sie das Gerät wahrscheinlich an ein Ladegerät anschließen, wenn der Akku mitten in der Arbeit leer wird. Haben Sie zu Hause oder in Ihrer Wohnung Ersatzglühbirnen für den Fall, dass eine Birne ausfällt? Wie sieht es mit einer Taschenlampe oder Kerzen für einen Stromausfall aus?

Die meisten Menschen sorgen für eine gewisse Redundanz oder haben einen Notfallplan, sowo hl im täglichen Leben als auch insbesondere in der IT. Wer auf einen Ersatzreifen oder eine Ersatzglühbirne zurückgreifen kann, muss bei einem Ausfall oder Defekt eine nur minimale Unterbrechung bewältigen. Wenn Sie Ihre Anwendungen für Redundanz entwerfen und entwickeln, bieten Sie Ihren Kunden ein hohes Maß an Verfügbarkeit, die alle Unterbrechungen der Anwendung minimiert oder sogar verbirgt. Alle Azure-Rechenzentren sind für Hochverfügbarkeit ausgelegt. Notstromversorgung, mehrere Netzwerkverbindungen und Speicher-Arrays mit Ersatzplatten sind nur einige der zentralen Redundanzkonzepte, die Azure für Sie bereitstellt und verwaltet. Aber die gesamte von Azure bereitgestellte Redundanz hilft Ihnen nicht, wenn Sie Ihre Anwendung auf einer einzigen VM ausführen. Um Ihnen Flexibilität und Kontrolle darüber zu geben, wie Sie Ihre Anwendung hochverfügbar machen, stehen Ihnen zwei Hauptmerkmale für IaaS-Workloads zur Verfügung:

- *Verfügbarkeitszonen* – damit können Sie VMs über physisch isolierte Segmente einer Azure-Region verteilen, um Ihre Anwendungsredundanz noch weiter zu maximieren. Verfügbarkeitszonen können auch für eine hohe Verfügbarkeit von Netzwerkressourcen sorgen, z. B. für öffentliche IP-Adressen und Load Balancer.
- *Verfügbarkeitsgruppen* – damit können Sie VMs logisch gruppieren, um sie über ein einzelnes Azure-Rechenzentrum zu verteilen und Störungen durch Ausfälle oder Wartungsupdates zu minimiere.

Ich rate Ihnen, für die meisten Bereitstellungen neuer Anwendungen in Azure, die Verwendung von Verfügbarkeitszonen zu planen. Dieser Ansatz bietet Flexibilität bei der Verteilung Ihrer Anwendung und bietet Redundanz für die Netzwerkressourcen, die oft von zentraler Bedeutung dafür sind, wie Kund*innen letztlich auf die zugrunde liegenden VMs zugreifen. Sehen wir uns diese Ansätze genauer an, um ihre Funktionsweise zu verstehen.

7.2 Infrastrukturredundanz mit Verfügbarkeitszonen

Verfügbarkeitszonen sind physisch getrennte Rechenzentren, die mit unabhängigen zentralen Betriebsmitteln arbeiten, z. B. für die Stromversorgung und die Netzwerkanbindung. Alle Azure-Regionen, die Verfügbarkeitszonen unterstützen, bieten drei Zonen. Sie erstellen Ihre Ressourcen innerhalb dieser Zonen und darüber herausreichend. Abbildung 7.2 zeigt, wie Azure-Ressourcen über Verfügbarkeitszonen hinweg verteilt werden können.

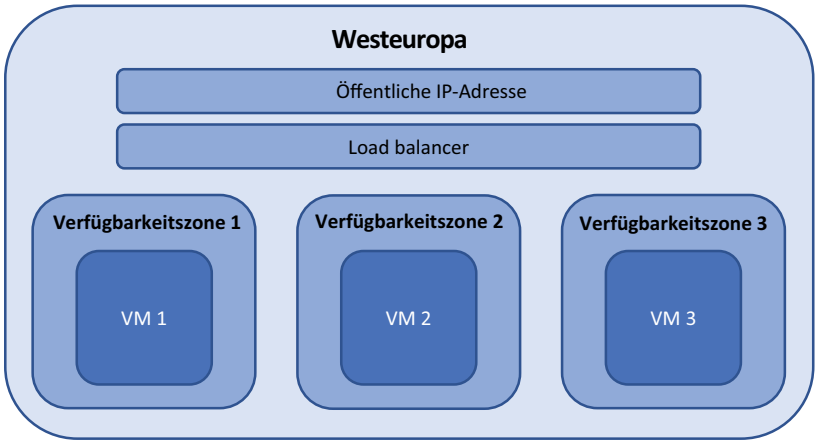


Abbildung 7.2 Eine Azure-Region kann mehrere Verfügbarkeitszonen enthalten: Physisch isolierte Rechenzentren, die im Hinblick auf Stromversorgung, Networking und Kühlung unabhängig sind. Virtuelle Azure-Netzwerkressourcen, wie öffentliche IP-Adressen und Load Balancer, können über alle Zonen in einer Region hinweg reichen und für mehr als nur für die VMs Redundanz bieten.

Mit den Verfügbarkeitszonen verkraften es Ihre Anwendungen, wenn ein ganzes Azure-Rechenzentrum offline geht. Damit diese Situation eintritt, müsste sich natürlich ein schwerwiegender Vorfall ereignen, der ist aber nicht unmöglich!

In großen Anwendungsbereitstellungen können Sie in jeder Verfügbarkeitszone mehr als eine VM erstellen. Mehrere VMs in einer Verfügbarkeitszone werden automatisch auf die verfügbare Hardware innerhalb der Zone verteilt. Es gibt nichts, was Sie konfigurieren müssen oder kontrollieren können. Selbst wenn sich ein Wartungsupdate oder ein Geräteausfall innerhalb einer Zone auf all Ihre VMs auswirken würde, die in der Zone laufen, sind Zonen physisch voneinander isoliert – Ihre VMs in einer anderen Zone würden also weiterlaufen.

Falls Sie damit nicht besonders glücklich sind: Könnten Ihre VMs in verschiedenen Zonen alle gleichzeitig von Wartungsupdates betroffen sein? Ja, das ist aber unwahrscheinlich. Für Zonen innerhalb einer Region gelten gestaffelte Updatezyklen. Updates werden in einer Zone durchgeführt. Sobald sie abgeschlossen sind, werden Updates in der nächsten Zone durchgeführt. Verfügbarkeitszonen bieten ein hohes Maß an Abstraktion und Redundanz und Sie müssen sich Ihre Anwendung über die gesamte Bereitstellung hinweg ansehen, nicht nur, wo sich VMs in einer Zone befinden.

Die Einbeziehung der virtuellen Netzwerkressourcen in die Verfügbarkeitszonen ist weitaus wichtiger, als es auf den ersten Blick scheinen mag. Abbildung 7.3 zeigt, was passieren würde, wenn das Rechenzentrum für Netzwerkressourcen, wie eine über

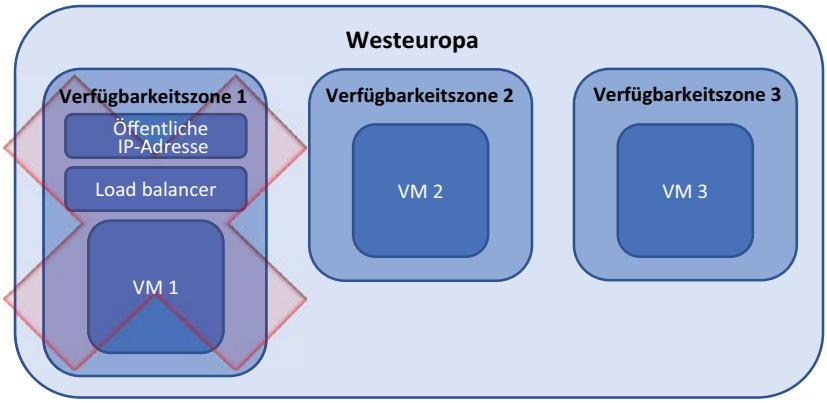


Abbildung 7.3 Wenn Netzwerkressourcen mit einem einzigen Azure-Rechenzentrum oder einer Zone verbunden sind, führt ein Ausfall in dieser Einrichtung dazu, dass die gesamte Anwendung für den Kunden nicht mehr erreichbar ist. Dabei spielt es keine Rolle, dass die anderen VMs in anderen Zonen weiterhin laufen. Ohne Netzwerkverbindung, über die der von Ihren Kund*innen kommende Datenverkehr verteilt wird, ist die gesamte Anwendung nicht erreichbar.

Verfügbarkeitszonen hinweg laufende öffentliche IP-Adresse und einen Load Balancer, nicht mehr verfügbar wäre.

Ich werde in Kapitel 8 eingehender auf Load Balancer eingehen. Im Moment müssen Sie nur verstehen, dass der Load Balancer den Datenverkehr über alle mit ihm verbundenen verfügbaren VMs verteilt. Die VMs melden in festgelegten Abständen ihren Integritätsstatus. Der Load Balancer stellt die Verteilung des Datenverkehrs an eine VM ein, sobald diese sich als nicht verfügbar meldet. Bei einem Load Balancer, der über Verfügbarkeitszonen hinweg arbeitet, führt ein Ausfall in einem Azure-Rechenzentrum dazu, dass dessen VMs nicht mehr verfügbar sind und aus der Lastausgleichsrotation herausgenommen werden.

Eine öffentliche IP-Adresse, die über die Verfügbarkeitszonen hinweg reicht, bietet Kund*innen einen einzigen Einstiegspunkt, über den sie den Load Balancer erreichen können, um an eine verfügbare VM verteilt zu werden. In einer Anwendungsbereitstellung, in der sich diese öffentliche IP-Adresse in einem einzigen Azure-Rechenzentrum befindet, kann kein*e Kund*in mehr auf die öffentliche IP-Adresse zugreifen, sobald in diesem Rechenzentrum ein Problem auftritt. Der oder die Kund*in kann Ihre Anwendung nicht benutzen, selbst wenn VMs verfügbar sind, um Kundenanfragen zu bearbeiten.

Ressourcen, die Verfügbarkeitszonen verwenden können, umfassen sowohl Zonendienste als auch zonenredundante Dienste:

- *Zonendienste* sind für Dinge wie VMs, eine öffentliche IP-Adresse oder ein Load Balancer. Die gesamte Ressource selbst wird innerhalb einer bestimmten Zone ausgeführt und kann selbsttätig sein, wenn eine andere Zone nicht verfügbar ist.
- *Zonenredundante Dienste* sind für Ressourcen, die automatisch zonenübergreifend repliziert werden können, wie zonenredundanter Speicher und SQL-Datenbanken. Die gesamte Ressource wird nicht innerhalb einer bestimmten Zone ausgeführt, Vielmehr werden ihre Daten auf mehrere Zonen verteilt, sodass sie auch dann noch verfügbar ist, wenn in einer Zone ein Problem auftritt.

Unterstützung für Verfügbarkeitszonen ist für mehr als 20 Azure-Dienste in mehr als zehn Regionen verfügbar. Die Anzahl der Dienste und Regionen, die in Verfügbarkeitszonen integriert werden, wächst weiter. Angesichts der regionalen Einschränkungen kann es jedoch vorkommen, dass die Unterstützung von Verfügbarkeitszonen für Kernressourcen wie VMs nicht verfügbar ist. In diesen Fällen gibt es eine andere Art von VM-Redundanz, die Sie in jeder Region verwenden können. Darauf gehen wir in Abschnitt 7.2.1: Verfügbarkeitsgruppen ein.

7.2.1 Erstellen von Netzwerkressourcen über eine Verfügbarkeitszone hinweg

Um einige dieser Verfügbarkeits- und Redundanzfunktionen in Aktion zu sehen, erstellen wir einige gemeinsame Ressourcen, wie z. B. eine öffentliche IP-Adresse und einen Load Balancer, und dann VMs. Das Ziel ist es, zu sehen, dass Sie nicht viel konfigurieren müssen, um die Vorteile von Verfügbarkeitszonen in Azure zu nutzen. Dies sind einfache Beispiele, sie bilden aber den Kern der meisten Anwendungsumgebungen, die Sie bereitstellen würden.

Öffentliche IP-Adressen und Load Balancers können in einer von zwei verfügbaren Versionen erstellt werden: Basic und Standard. Der primäre Unterschied besteht darin, dass die Netzwerkressource des Standardtyps Verfügbarkeitszonen nutzen kann. In der Standardversion sind eine öffentliche IP-Adresse oder ein Load Balancer automatisch zonenredundant. Eine weitere Konfiguration gibt es nicht. Azure Platform speichert die Metadaten für die Ressource in der von Ihnen angegebenen Region zentral und stellt sicher, dass die Ressource weiter ausgeführt wird, falls eine Zone nicht mehr verfügbar ist.

Machen Sie sich im Moment nicht zu viele Gedanken darüber, was mit dem Load Balancer und den Netzwerkressourcen passiert. Denken Sie an das, was ich zu Beginn gesagt habe – die nächsten zwei oder drei Kapitel bauen alle aufeinander auf. In Kapitel 8 sehen wir uns Load Balancer im Detail an. Dann wird all dies verständlicher.

Jetzt testen

Führen Sie die folgenden Schritte aus, um über Verfügbarkeitszonen hinweg redundante Netzwerkressourcen zu erstellen:

1. Wählen Sie oben im Dashboard des Azure Portals das Cloud Shell-Symbol.
2. Erstellen Sie eine Ressourcengruppe, z. B. `azuremolchapter7az`:

```
az group create --name azuremolchapter7az --location westeurope
```

3. Erstellen Sie in Ihrer Ressourcengruppe eine öffentliche IP-Adresse in der Standardversion. Standardmäßig würde eine öffentliche IP-Adresse vom Typ *Basic* erstellt und nur einer einzigen Zone zugeordnet. Der Parameter `--sku standard` weist Azure an, eine redundante, zonenübergreifende Ressource zu erstellen:

```
az network public-ip create \
  --resource-group azuremolchapter7az \
  --name azpublicip \
  --sku standard
```

4 Erstellen Sie einen Load Balancer, der sich über Verfügbarkeitszonen hinweg erstreckt. Auch hier würde standardmäßig ein Load Balancer vom Typ Basic erstellt und einer einzigen Zone zugeordnet. Das ist aber nicht der Hochverfügbarkeitsansatz, den Ihre Anwendungen besitzen sollen. Legen Sie für das Laden der SKU wie folgt *standard* fest, um einen zonenredundanten Load Balancer zu erstellen:

```
az network lb create \
  --resource-group azuremolchapter7az \
  --name azloadbalancer \
  --public-ip-address azpublicip \
  --sku standard
```

7.2.2 Erstellen von VMs in einer Verfügbarkeitszone

Um eine VM in einer Verfügbarkeitszone zu erstellen, legen Sie fest, in welcher Zone die VM ausgeführt werden soll. Um viele VMs bereitzustellen, sollten Sie idealerweise eine Vorlage erstellen und verwenden. Die Vorlage definiert die Zonen für die einzelnen VMs und verteilt diese. Wenn die Nachfrage der Kund*innen in Ihrer Online-Pizzeria wächst, können Sie die Vorlage auch mit der dann gewünschten Anzahl VMs aktualisieren und erneut bereitstellen. Die neuen VMs werden automatisch über die Zonen verteilt. Sie müssen nicht manuell verfolgen, in welchen Zonen die VMs laufen. In der Übung am Ende des Kapitels werden Sie eine Vorlage verwenden, um automatisch mehrere VMs zu erstellen und zu verteilen. Lassen Sie uns eine VM erstellen und die Zone manuell festlegen, um den logischen Ablauf zu betrachten, in dem eine Zone für eine VM angegeben wird.

Jetzt testen

Führen Sie folgende Schritte aus, um eine VM in einer Verfügbarkeitszone zu erstellen:

1 Klicken Sie im Azure Portal oben im Dashboard auf das Cloud Shell-Symbol.

2 Erstellen Sie eine VM mit dem Befehl `az vm create`, den Sie bereits in früheren Kapiteln verwendet haben. Benutzen Sie den Parameter `--zone`, um festzulegen, dass die VM in der Zone 1, 2 oder 3 ausgeführt werden soll. Im folgenden Beispiel wird eine VM mit dem Namen `zonedvm` in Zone 3 erstellt:

```
az vm create \
  --resource-group azuremolchapter7az \
  --name zonedvm \
  --image ubuntults \
  --size Standard_B1ms \
  --admin-username azuremol \
  --generate-ssh-keys \
  --zone 3
```

Es dauert einige Minuten, um die VM zu erstellen. Nach Abschluss des Prozesses gibt der Befehl die Zone aus, in der die VM läuft. Diese Information können Sie auch mit dem Befehl `az vm show` anzeigen:

```
az vm show \
--resource-group azuremolchapter7az \
--name zonedvm \
--query zones
```

HINWEIS Die in dieser praktischen Übung verwendeten Beispiele sind zwar einfach, sie zeigen aber auf, dass Zonen nur wenig Konfiguration benötigen. Sie haben keinen zonenredundanten Load Balancer und keine VM integriert. In Kapitel 8 werden Sie aber eine nutzbarere Anwendungsumgebung erstellen, die über Verfügbarkeitszonen hinweg verteilt ist. Hier wollen wir nur zeigen, dass die Azure-Plattform die Redundanz und die Verteilung Ihrer Ressourcen übernimmt, sodass Sie sich auf die Anwendung selbst konzentrieren können.

7.3 VM-Redundanz mit Verfügbarkeitsgruppen

Verfügbarkeitszonen eignen sich hervorragend für die Entwicklung von Redundanz für eine größere Anzahl von Ressourcen, die Ihre Anwendungen und Workloads ausmachen. Ich empfehle Ihnen, nach Möglichkeit diese für neue Workloads zu verwenden. Manchmal müssen Sie jedoch nicht unbedingt alle Ressourcenzone redundant machen. Vielleicht möchten Sie auch VMs in einer Azure-Region erstellen, die derzeit keine Verfügbarkeitszonen unterstützt.

Wenn Sie nur Redundanz für VMs bereitstellen möchten, reichen Verfügbarkeitsgruppen völlig aus. Diese sind bewährt, zuverlässig und in allen Regionen verfügbar. Verfügbarkeitsgruppen enthalten eine logische Gruppe von VMs, die der Azure-Plattform anzeigen, dass die zugrunde liegende Hardware, auf der diese VMs laufen, sorgfältig ausgewählt werden muss. Wenn Sie zwei VMs erstellen, die auf demselben physischen Server laufen, dann fallen beide VMs aus, sobald der Server ausfällt. Bei Zehntausenden oder mehr physischen Servern in einem Azure-Rechenzentrum ist es sehr unwahrscheinlich, dass Ihre beiden VMs auf demselben Server laufen, es ist aber möglich! Es muss kein Fehler auftreten, aber vielleicht sorgt ein Wartungsupdate dafür, dass der physische Server für kurze Zeit nicht verfügbar ist.

Wie sieht es aus, wenn Ihre VMs in demselben Rack laufen und mit derselben Speicher- oder Netzwerkausrüstung verbunden sind? Sie haben erneut den Single Point of Failure, über den ich zu Beginn des Kapitels gesprochen habe.

Verfügbarkeitsgruppen erlauben es der Azure-Plattform, Ihre VMs über logische Gruppen hinweg zu erstellen, die als *Fehlerdomänen* und *Update-Domänen* bezeichnet werden. Über diese logischen Domänen kann Azure Platform die physischen Grenzen von Hardwaregruppen nachvollziehen, um sicherzustellen, dass Ihre VMs darüber gleichmäßig verteilt sind. Wenn mit einer Hardware ein Problem auftritt, sind davon nur wenige VMs in Ihrer Verfügbarkeitsgruppe betroffen. Oder wenn auf die physische Hardware Wartungsupdates angewendet werden, dann wirkt sich die Wartung nur auf einige Ihrer VMs aus. Die Beziehung zwischen physischer Hardware und logischen Fehler- und Update-Domänen innerhalb einer Verfügbarkeitsgruppe wird in Abbildung 7.4 dargestellt.

Verfügbarkeitszonen führen dieselbe Art der Verteilung durch, aber sie ist abstrahiert und wird nicht offengelegt. Selbst bei Verfügbarkeitsgruppen können Sie nicht viel konfigurieren. Aber es ist hilfreich zu wissen, was hinter den Kulissen passiert.

7.3.1 Fehlerdomänen

Eine *Fehlerdomäne* ist eine logische Gruppe von Hardware in einem Azure-Rechenzentrum. Sie enthält Hardware, die gemeinsame Stromversorgungs- oder Netzwerkgeräte nutzt. Über diese Fehlerdomänen haben Sie keine Kontrolle, und es ist keine Konfiguration auf der VM-Ebene möglich. Die Azure-Plattform verfolgt, in welchen Fehlerdomänen Ihre VMs platziert sind und verteilt neue VMs so über diese

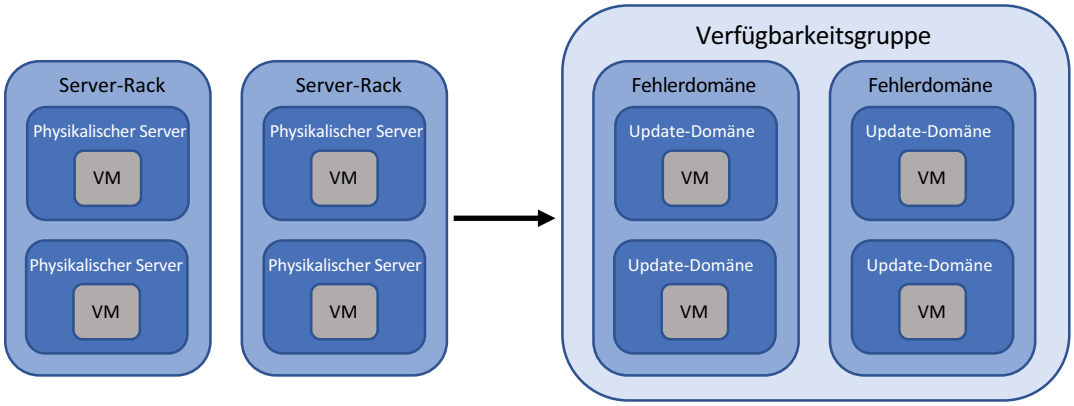


Abbildung 7.4 In einem Azure-Rechenzentrum ist die Hardware logisch in Update-Domänen und Fehlerdomänen unterteilt. Anhand dieser logischen Domänen kann Azure Platform nachvollziehen, wie Ihre VMs über die zugrunde liegende Hardware verteilt werden müssen, um Ihre Redundanzanforderungen zu erfüllen. Dies ist ein einfaches Beispiel – eine Update-Domäne enthält wahrscheinlich mehr als einen physischen Server.

Fehlerdomänen, dass Ihnen immer VMs zur Verfügung stehen, falls der Strom oder ein Netzwerkswitch ausfallen.

VMs, die verwaltete Datenträger verwenden (denken Sie daran, dass *alle* Ihre VMs verwaltete Datenträger verwenden sollten!), berücksichtigen außerdem die logischen Grenzen und die Verteilung von Fehlerdomänen. Azure Platform führt eine logische Zuweisung von Speicherclustern zu Fehlerdomänen durch, um sicherzustellen, dass Ihre VMs über Hardwaregruppen verteilt werden. Auch die verwalteten Datenträger sind über Speicherhardware hinweg verteilt. Es würde für die VM-Redundanz über Serverhandware hinweg keinen Sinn ergeben, wenn es möglich wäre, dass sich alle verwalteten Datenträger letztlich in einem Speichercluster befänden! Und ja, verwaltete Datenträger können auch mit Verfügbarkeitszonen verwendet werden.

7.3.2 Update-Domänen

Während Fehlerdomänen eine logische Hardwaregruppe bilden, um vor Hardwareausfällen zu schützen, sorgen Update-Domänen für den Schutz vor Ausfällen wegen routinemäßiger Wartung. Um diesen Schutz zu bieten, wird eine Fehlerdomäne logisch weiter in Update-Domänen unterteilt. Auch hier müssen Sie nichts konfigurieren. Über Update-Domänen kann Azure Platform nachvollziehen, wie Ihre VMs über Ihre Verfügbarkeitsgruppe verteilt werden müssen.

Azure-Techniker*innen führen die (meist automatisierten) Wartungsarbeiten und Updates für die gesamte physische Hardware in einer Update-Domäne durch. Anschließend wird die gleiche Wartung für die gesamte Hardware in der nächsten Update-Domäne durchgeführt. Diese Wartungsarbeiten werden an den Update-Domänen zeitlich versetzt ausgeführt, um sicherzustellen, dass nicht alle VMs in einer Verfügbarkeitsgruppe auf Hardware laufen, die gleichzeitig gewartet wird. Es ist der gleiche Prozess wie bei den Verfügbarkeitszonen. Die Verteilung Ihrer Ressourcen bedeutet, dass Sie kein Szenario haben können, in dem die gesamte zugrunde liegende Hardware für Ihre Ressourcen zur gleichen Zeit aktualisiert wird.

Zwischen Domänen bestehen keine Beziehungen über mehrere Verfügbarkeitsgruppen hinweg. Die physischen Ressourcen, aus denen Fehler- und Update-Domänen in einer Verfügbarkeitsgruppe bestehen, sehen in einer zweiten Verfügbarkeitsgruppe möglicherweise anders aus. Wenn Sie also mehrere Verfügbarkeitsgruppen erstellen und Ihre VMs auf diese verteilen, dann enthält beispielsweise Fehlerdomäne 1 nicht immer dieselbe physische Hardware.

7.3.3 Verteilen von VMs in einer Verfügbarkeitsgruppe

Gehen wir schrittweise vor und sehen uns an, wie VMs über die logischen Fehler- und Update-Domänen verteilt werden, die eine Verfügbarkeitsgruppe bilden. Auf diese Weise haben Sie mehrere VMs, auf denen Sie Ihre Pizzeria betreiben können, und kein*e Kund*in muss hungern!

Jetzt testen

Führen Sie die folgenden Schritte aus, um eine Resource Manager-Vorlage bereitzustellen und die Verfügbarkeitsgruppen in Aktion zu sehen:

1. Öffnen Sie in einem Webbrowser eine Resource Manager-Vorlage aus dem GitHub-Repository unter https://github.com/fouldsy/azure-mol-samples-2nd-ed/tree/master/ 07/availability-set, und klicken Sie auf die Schaltfläche „Deploy to Azure" (Bereitstellung in Azure). In dieser Übung verwenden Sie eine Vorlage, mit der Sie VMs schnell bereitstellen können, um dann zu untersuchen, wie diese VMs über Verfügbarkeitsgruppen verteilt sind.

   Das Azure Portal wird geöffnet und fordert einige Parameter an.

2. Erstellen Sie eine neue Ressourcengruppe, und geben Sie einen Namen ein, beispielsweise `azuremolchapter7`. Wählen Sie eine Region aus und geben Sie Ihre SSH-Schlüsseldaten an (diese können Sie in der Cloud Shell mit cat ~/.ssh/id_rsa.pub abrufen).

   Die Vorlage erstellt eine Verfügbarkeitsgruppe, die drei VMs enthält. Diese VMs werden auf die logischen Fehler- und Update-Domänen verteilt. Aufbauend auf dem, was Sie in Kapitel 6 über den Resource Manager gelernt haben, verwendet diese Vorlage die Funktion `copyIndex()`, um mehrere VMs und NICs zu erstellen.

3 Aktivieren Sie das Kontrollkästchen „I agree to the terms and conditions stated above" (Ich stimme den oben genannten Geschäftsbedingungen zu), um zu bestätigen, dass Sie die in der Vorlage beschriebenen Ressourcen erstellen möchten, und klicken Sie dann auf „Purchase" (Kaufen).

Es dauert ein paar Minuten, bis alle drei VMs in der Verfügbarkeitsgruppe erstellt sind. Lesen Sie sich den Rest dieses Abschnitts durch, während die Bereitstellung im Portal fortgesetzt wird.

Wenn die Bereitstellung der Vorlage beginnt, wird eine Verfügbarkeitsgruppe erstellt und die Anzahl der von Ihnen gewünschten Update- und Fehlerdomänen wird zugewiesen. In der Beispielvorlage wurden folgende Eigenschaften definiert:

```
"properties": {
    "platformFaultDomainCount": "2",
    "platformUpdateDomainCount": "5",
    "managed": "true"
}
```

Diese Eigenschaften erstellen eine Verfügbarkeitsgruppe mit zwei Fehlerdomänen und fünf Updated-Dmänen, wie in Abbildung 7.5 dargestellt. Sie geben außerdem an, dass die VMs verwaltete Datenträger verwenden, um auch eine Verteilung der Datenträger sicherzustellen. Die von Ihnen für die Verfügbarkeitsgruppe ausgewählte Region bestimmt die maximale Anzahl von Fehler- und Update-Domänen. Regionen unterstützen entweder 2 oder 3 Fehlerdomänen und bis zu 20 Update-Domänen.

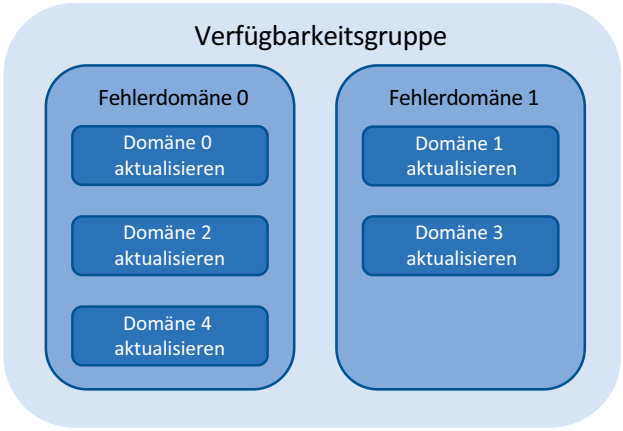


Abbildung 7.5 Die Verfügbarkeitsgruppe, die Ihre Beispielvorlage bereitstellt, enthält zwei Fehlerdomänen und fünf Update-Domänen. Die Nummerierung beginnt mit Null. Die Update-Domänen werden sequenziell über die Fehlerdomänen hinweg erstellt.

Wenn Sie mehr VMs in einer Verfügbarkeitsgruppe erstellen, müssen Sie sich überlegen, wie viele Update-Domänen verwendet werden sollen. Wenn Sie z. B. fünf Update-Domänen erstellen, bedeutet dies, dass bis zu 20 % Ihrer VMs aufgrund von Wartungsarbeiten möglicherweise nicht verfügbar sind:

- Angenommen, in Ihrer Verfügbarkeitsgruppe befinden sich zehn VMs. Dann bedeutet dies, dass zwei dieser VMs gleichzeitig gewartet werden können. Wenn Sie möchten, dass immer nur eine VM gewartet wird, müssen Sie zehn Update-Domänen erstellen. Je mehr Update-Domänen Sie erstellen, desto länger befindet sich Ihre Anwendung potenziell in einem Wartungszustand.
- Fahren wir mit dem vorherigen Beispiel von zehn VMs in zehn Update-Domänen fort. Jetzt kann es zu Unterbrechungen bei Ihren Anwendungen kommen, bis alle zehn dieser Update-Domänen ihren Wartungszyklus abgeschlossen haben. Wenn Sie nur fünf Updatedomains haben, verkürzt sich dieser Wartungszeitraum. Es ist nicht unbedingt schlecht, eine längere Wartungszeit zu haben. Es kommt eher darauf an, wie groß Ihre Toleranz ist, möglicherweise mit weniger als der vollen Kapazität zu arbeiten.

Es ist wichtig, daran zu denken, dass diese Update-Domänen und Wartungszyklen von der Azure-Plattform selbst durchgeführt werden. Sie müssen auch Ihre eigenen Update-Anforderungen und Wartungsfenster berücksichtigen.

Wenn die erste VM erstellt wird, sucht die Azure-Plattform nach der ersten verfügbaren Bereitstellungsposition. Dies sind Fehlerdomäne 0 und Update-Domäne 0, wie in Abbildung 7.6 dargestellt.

Wenn die zweite VM erstellt wird, sucht die Azure-Plattform nach der nächsten verfügbaren Bereitstellungsposition. Dies sind jetzt Fehlerdomäne 1 und Update-Domäne 1, wie in Abbildung 7.7 dargestellt.

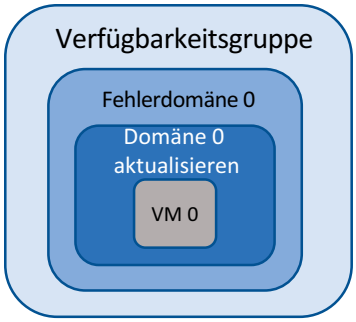


Abbildung 7.6 Die erste VM wird in Fehlerdomäne 0 und Update-Domäne 0 erstellt.

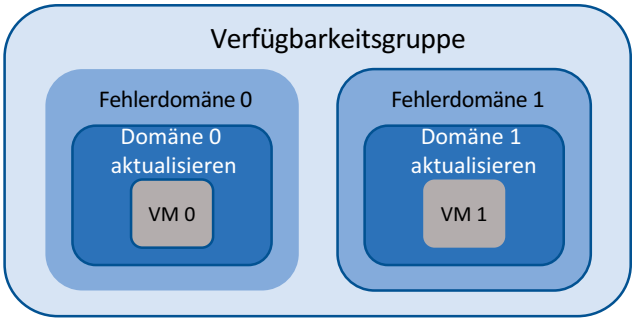


Abbildung 7.7 Nachdem die zweite VM erstellt wurde, sind die VMs jetzt gleichmäßig auf Fehler- und Update-Domänen verteilt. Dies wird meist als der minimale Redundanzumfang erachtet, der Ihre Anwendungen schützt.

Ihre Vorlage erstellt drei VMs, was glauben Sie, was als nächstes passiert? Azure Platform sucht erneut nach der nächsten verfügbaren Bereitstellungsposition. Sie haben nur zwei Fehlerdomänen erstellt, deshalb wird die VM erneut in der Fehlerdomäne 0 erstellt. Die VM wird aber in einer anderen Update-Domäne erstellt als die erste VM. Die dritte VM wird in Update-Domäne 2 erstellt, wie in Abbildung 7.8 dargestellt.

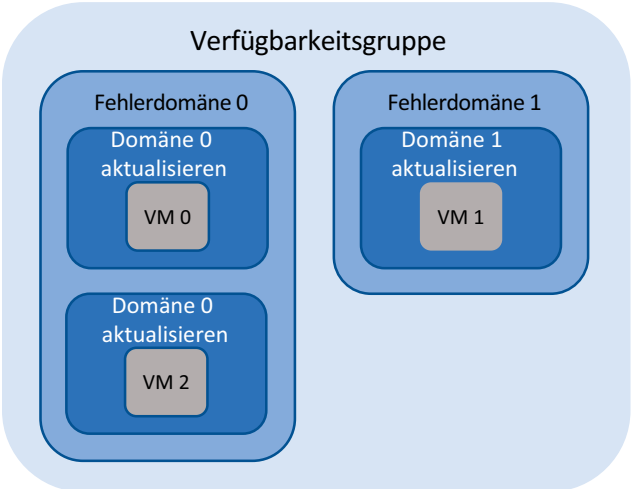


Abbildung 7.8 Die dritte VM wird wieder in Fehlerdomäne 0 erstellt, aber in Update-Domäne 2. Auch wenn für die VMs 0 und 2 möglicherweise dasselbe Risiko für einen Hardwareausfall besteht, befinden sie sich in verschiedenen Update-Domänen und werden daher nicht gleichzeitig einer regelmäßigen Wartung unterzogen.

Die VMs 0 und 2 befinden sich in derselben Fehlerdomäne, ein Hardwareausfall würde daher beide VMs betreffen. Die routinemäßige Wartung wirkt sich aber immer nur auf eine dieser VMs aus, da sie über die Update-Domänen hinweg verteilt sind. Wenn Sie fortfahren und weitere VMs erstellen, wird Azure Platform diese weiterhin über unterschiedliche Fehler- und Update-Domänen hinweg verteilen. Nachdem alle fünf Update-Domänen verwendet wurden, wird die sechste VM wieder in Update-Domäne 0 erstellt und der Kreislauf beginnt von vorne.

7.3.4 Anzeigen der Verteilung von VMs in einer Verfügbarkeitsgruppe

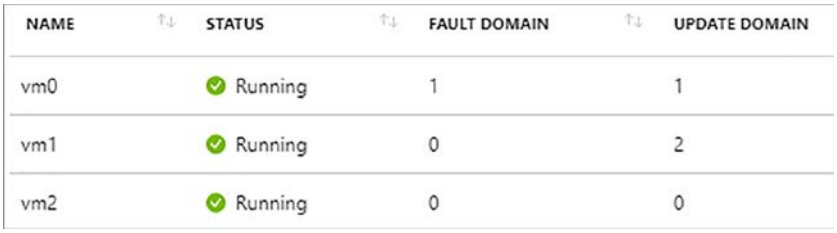
Da Sie jetzt wissen, wie VMs theoretisch auf Fehler- und Update-Domänen in einer Verfügbarkeitsgruppe verteilt werden, wollen wir prüfen, was bei der Bereitstellung Ihrer Resource Manager-Vorlage passiert ist.

Jetzt testen

Führen Sie die folgenden Schritte aus, um zu sehen, wie Ihre VMs in einer Verfügbarkeitsgruppe verteilt wurden:

1. Suchen Sie in der Navigationsleiste links im Azure Portal nach „Resource Group" (Ressourcengruppe), und wählen Sie diese Option aus.
2. Wählen Sie die Ressourcengruppe aus, die mit Ihrer Vorlage erstellt wurde, z. B. `azuremolchapter7`.
3. Wählen Sie Ihre Verfügbarkeitsgruppe in der Liste der Ressourcen aus, z. B. `azuremolavail abilityset`.

   Im Übersichtsfenster sehen Sie eine Liste der VMs mit den zugehörigen Fehler- und Update-Domänen, wie in Abbildung 7.9 dargestellt.



NAME	STATUS	FAULT DOMAIN	UPDATE DOMAIN
vm0	Running	1	1
vm1	Running	0	2
vm2	Running	0	0

Abbildung 7.9 Sie zeigt die in der Verfügbarkeitsgruppe enthaltenen VMs mit ihren jeweiligen Fehler- und Update-Domänen. In dieser Tabelle sehen Sie, wie die VMs über die logischen Domänen hinweg verteilt sind.

Wenn Sie genau aufpassen, können Sie feststellen, dass die VMs nicht exakt in der erwarteten Reihenfolge in den Fehler- und Update-Domänen auftauchen. Ist das ein Fehler? Wahrscheinlich nicht. Wenn Sie sich das Beispiel in Abbildung 7.9 ansehen und mit dem Konzept vergleichen, erwarten Sie, dass die VMs wie in Tabelle 7.1 verteilt werden.

Tabelle 7.1 Verfügbarkeitsgruppen-VMs, die sequenziell erstellt und über Domänen verteilt sind

Name	Fault domain	Update domain
vm0	0	0
vm1	1	1
vm2	0	2

Was ist also falsch gelaufen? Nichts. Denken Sie daran, wie Resource Manager Ressourcen aus einer Vorlage erstellt. Azure Platform wartet nicht, bis die erste VM erstellt wurde, bevor die zweite erstellt wird. Alle drei VMs werden gleichzeitig erstellt. Daher entscheiden Sekundenbruchteile darüber, welche VM einer Verfügbarkeitsgruppe zuerst zugordnet wird. Diese Reihenfolge spielt keine Rolle, weil Sie keine Kontrolle darüber haben, wofür die zugrunde liegenden Fehler- und Update-Domänen stehen. Darum kümmert sich ausschließlich die Azure-Plattform. Sie müssen nur sicherstellen, dass Ihre VMs verteilt *werden*, nicht *wohin*.

Ich möchte aber schöne Zahlen haben

Wenn Sie das serielle Erstellen von VMs stört und Sie die VMs in einer sauberen Reihenfolge verteilen *müssen*, können Sie Resource Manager anweisen, VMs in *serieller Reihenfolge* statt *parallel* zu erstellen. In diesem Modus werden die VMs nacheinander erstellt, weshalb sich die Bereitstellungszeit verlängert. Um dieses serielle Verhalten zu aktivieren, müssen Sie in Ihren Vorlagen in der `copyIndex()`-Funktion `"mode": "serial"` angeben. Das sollte die VMs auf saubere, sequenzielle Weise verteilen!

7.4 Übung: Bereitstellen von hochverfügbaren VMs mit einer Vorlage

Diese Übung kombiniert das, was Sie in Kapitel 6 über Azure Resource Manager und Vorlagen gelernt haben, mit Verfügbarkeitszonen. Nehmen Sie sich etwas Zeit und sehen Sie sich das Beispiel für die Schnellstartvorlage in dieser Übung an. Dort sehen Sie, wie Sie Logik und Funktionen nutzen können, um mehrere VMs über Zonen zu verteilen. Stellen Sie nicht einfach nur die Vorlage bereit – schauen Sie sich an, wie die Vorlage auf den in Kapitel 6 eingeführten Funktionen aufbaut!

Was ist ein Kontingent?

In Azure verhindern Standardkontingente für Ihr Abonnement, dass Sie versehentlich eine Vielzahl Ressourcen bereitstellen und diese dann vergessen, was Sie viel Geld kosten würde. Diese Kontingente variieren in der Regel nach Ressourcentyp und Abonnementtyp und werden auf der Ebene der Regionen durchgesetzt. Eine vollständige Liste der Kontingente finden Sie unter http://mng.bz/ddcx.

Wenn Sie in diesen nächsten Kapiteln beginnen werden, mehrere VMs zu erstellen, können Kontingentprobleme auftreten. Kontingentprobleme können auch dann auftreten, wenn Sie die Ressourcen aus früheren Kapiteln und Übungen nicht löschen. Kontingente bieten ein gutes System, das Sie über Ihren Ressourcenverbrauch informiert. Die Fehlermeldungen sind vielleicht schwer verständlich. Wenn Sie aber eine Fehlermeldung wie

```
Operation results in exceeding quota limits of Core.
Maximum allowed: 4, Current in use: 4, Additional requested: 2.
```

sehen, ist dies ein sicheres Zeichen dafür, dass Sie eine Erhöhung Ihrer Kontingente beantragen müssen. Das ist nicht kompliziert und nichts, was nur für Azure gilt. Sie können Ihr aktuelles Kontingent für eine bestimmte Region wie folgt anzeigen:

```
az vm list-usage --location eastus
```

Wenn Sie Probleme mit dieser Übung haben, löschen Sie die beiden ersten Ressourcengruppen, die Sie in diesem Kapitel erstellt haben, d. h. `azuremolchapter7` und `azuremolchapter7az`. Wenn Sie ein niedriges Standardkontingent festgelegt haben, können die vier VMs in diesen Ressourcengruppen verhindern, dass Sie diese Übung erfolgreich abschließen können.

Führen Sie die unter http://mng.bz/Xq2f beschriebenen Schritte aus, um eine Erhöhung des Kontingents für eine Region anzufordern.

Sehen wir uns eine Beispielvorlage an, die mehrere VMs über Verfügbarkeitszonen hinweg bereitstellt.

1. Öffnen Sie im Webbrowser die JSON-Datei unter https://github.com/Azure/azure-quick-start-templates/blob/master/201-multi-vm-lb-zones/azuredeploy.json, und suchen Sie nach dem folgenden Text:

   ```
   Microsoft.Compute/virtualMachines
   ```

   Der Abschnitt für die VMs sieht ähnlich aus wie der in Kapitel 6. Achten Sie auf den Wert der Eigenschaft zones. In diesem Abschnitt werden einige Funktionen kombiniert, die in Vorlagen verfügbar sind, um beim Erstellen der VM entweder Zone 1, 2 oder 3 auszuwählen. Auf diese Weise müssen Sie nicht manuell verfolgen, welche VM in welcher Zone ausgeführt wird und wie Sie dann zusätzliche VMs bereitstellen müssen.
2. Suchen Sie im Webbrowser nach den folgenden Einträgen, um die Abschnitte für die öffentliche IP-Adresse und den Load Balancer anzuzeigen:

   ```
   Microsoft.Network/publicIPAddresses
   Microsoft.Network/loadBalancers
   ```

   Beide Ressourcen verwenden die Standard-SKU, was für standardmäßige Zonenredundanz sorgt. Dafür ist keine zusätzliche Konfiguration erforderlich! Sehen wir uns das in der Praxis an.

3 Öffnen Sie in Ihrem Webbrowser die Quickstart-Vorlage unter http://mng.bz/O69a, und klicken Sie auf die Schaltfläche „Deploy to Azure" (Bereitstellung in Azure).
4 Erstellen Sie eine Ressourcengruppe oder wählen Sie eine aus, und geben Sie den Benutzungsnamen und das Kennwort für die VMs an.
5 Geben Sie einen eindeutigen DNS-Namen ein, z. B. „azuremol".
6 Wählen Sie aus, ob sie Linux- oder Windows-VMs erstellen möchten. Das Erstellen von Windows-VMs dauert etwas länger.
7 Geben Sie an, wie viele VMs erstellt werden sollen, z. B. „3".
8 Aktivieren Sie das Kontrollkästchen, um die Bedingungen für die Bereitstellung der Vorlage zu akzeptieren, und klicken Sie dann auf „Purchase " (Kaufen), wie in Abbildung 7.10 dargestellt.

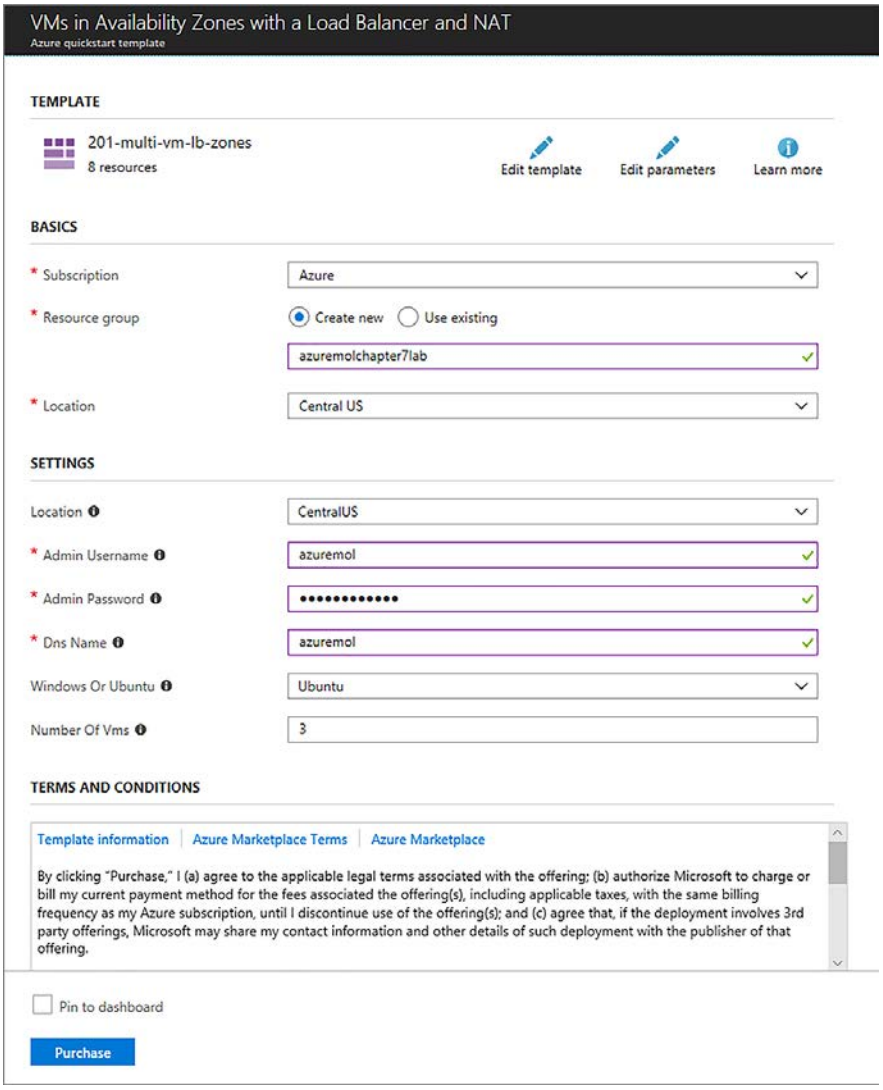


Abbildung 7.10 Um die Verfügbarkeitszonenvorlage im Azure Portal bereitzustellen, geben Sie eine Ressourcengruppe, den Benutzungsnamen und das Kennwort an und dann den Betriebssystemtyp und die Anzahl der VMs, die Sie erstellen möchten. Die Vorlage verwendet Schleifen, `copyIndex()`, `dependsOn`, Variablen und Parameter, die wir in Kapitel 6 über Resource Manager behandelt haben.

Benutzen Sie, nachdem die VMs erstellt wurden, das Azure Portal oder den Befehl `az vm show`, um anzuzeigen, wie die VMs über Zonen verteilt wurden. Falls Sie wissen möchten, was der Rest der Vorlage mit den Netzwerkressourcen anstellt, können Sie in Kapitel 8 die Details der Load Balancer erleben!

Putzkolonne

Erinnern Sie sich, dass ich zu Beginn des Buchs gesagt hatte, dass Sie unbedingt aufräumen müssen, um Ihre Azure-Kosten zu minimieren? Ich rate Ihnen dringend, die Ressourcengruppen zu löschen, die Sie in diesem Kapitel erstellt haben. In den nächsten Kapiteln werden wir weiterhin mehrere VMs und Web App-Instanzen erstellen. Behalten Sie daher Ihre Kosten und Kontingente im Auge.

Jedes Mal, wenn Sie sich beim Azure Portal anmelden, erhalten Sie eine Popup-Benachrichtigung, die Sie über Ihr Azure-Guthaben informiert. Wenn Sie sehen, dass sich Ihr verfügbares Guthaben von Tag zu Tag um einen großen Betrag verringert, prüfen Sie, ob Sie vergessen haben, Ressourcengruppen zu löschen!

8 Anwendungen mit Lastenausgleich

Ein wichtiger Bestandteil von hochverfügbaren Anwendungen ist die Verteilung des Datenverkehrs auf alle VMs. In Kapitel 7 haben Sie den Unterschied zwischen Verfügbarkeitsgruppen und Verfügbarkeitszonen kennengelernt und erfahren, wie Sie mehrere VMs über Azure-Rechenzentren oder -Regionen hinweg erstellen, um für Anwendungsredundanz zu sorgen. Aber all diese hochverfügbaren und verteilten VMs helfen Ihnen nicht, wenn nur eine VM den gesamten Kunden-Traffic erhält.

Load Balancer sind Netzwerkressourcen, die den von Ihren Kund*innen eingehenden Anwendungsdatenverkehr erhalten, den Datenverkehr prüfen, um Filter und Lastausgleichsregeln anzuwenden, und die Anfragen dann an einen Pool von VMs verteilen, die Ihre Anwendung ausführen. Bei Azure gibt es verschiedene Möglichkeiten, für einen Lastausgleich des Datenverkehrs zu sorgen, beispielsweise wenn Sie ein SSL-Offloading für große Anwendungen durchführen müssen, die einen verschlüsselten Netzwerkdatenverkehr verwenden. In diesem Kapitel erfahren Sie mehr über die verschiedenen Load Balancer-Komponenten, wie Sie Datenverkehrsregeln und Filter konfigurieren und wie Sie den Datenverkehr an VMs verteilen. Sie bauen auf den Hochverfügbarkeitskomponenten aus Kapitel 8 auf und bereiten sich auf das Kapitel 9 vor, in dem es darum geht, Ressourcen zu skalieren.

8.1 Azure Load Balancer-Komponenten

Load Balancer können bei Azure auf zwei Schichten arbeiten: Schicht 4 (die Transportschicht), in der nur der Netzwerkdatenverkehr geprüft und verteilt wird, und Schicht 7, auf der die Anwendungsdaten im Netzwerkdatenverkehr geprüft werden, um die Verteilung der Daten zu bestimmen. Load Balancer arbeiten in beiden Schichten auf die gleiche Weise, wie in Abbildung 8.1 dargestellt.

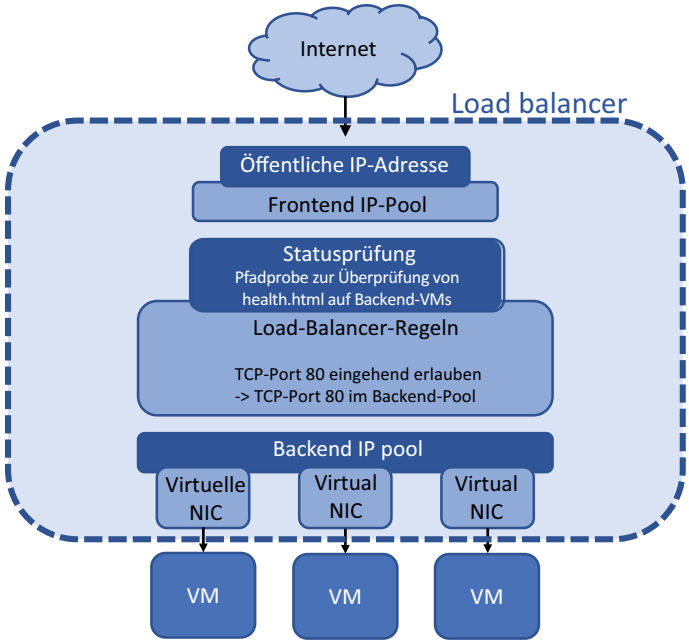


Abbildung 8.1 Datenverkehr aus dem Internet erreicht den Load Balancer über eine öffentliche IP-Adresse, die mit einem Frontend-IP-Pool verbunden ist. Der Datenverkehr wird durch Lastausgleichsregeln verarbeitet, die bestimmen, wie und wohin der Datenverkehr weitergeleitet wird. Mit den Regeln verbundene Integritätstests sorgen dafür, dass der Datenverkehr nur an fehlerfreie Knoten verteilt wird. Ein Backend-Pool virtueller NICs, die mit VMs verbunden sind, erhält dann den Datenverkehr, der durch die Load Balancer-Regeln verteilt wird.

Ein Load Balancer besteht aus wenigen Hauptkomponenten:

- *Frontend-IP-Pool* – der Einstiegspunkt für den Load Balancer. Um den Zugriff aus dem Internet zu ermöglichen, kann eine öffentliche IP-Adresse mit dem Frontend-IP-Pool verbunden werden. Für interne Load Balancer können private IP-Adressen bereitgestellt werden.
- *Integritätstests* – überwachen des Status der verbundenen VMs. Um sicherzustellen, dass der Datenverkehr nur an VMs verteilt wird, die fehlerfrei arbeiten und reagieren, werden regelmäßig Tests durchgeführt, um zu prüfen, ob eine VM ordnungsgemäß auf Datenverkehr reagiert.
- *Load Balancer-Regeln* – verteilen des Datenverkehrs an die VMs. Jedes eingehende Paket wird mit den Regeln verglichen, in denen eingehende Protokolle und Ports definiert sind, und dann an eine Reihe von zugeordneten VMs verteilt. Wenn keine Regel mit dem eingehenden Datenverkehr übereinstimmt, wird der Datenverkehr gelöscht.
- *Regeln für die Übersetzung von Netzwerkadressen (NAT)* – diese können Datenverkehr direkt an bestimmte VMs weiterleiten. Wenn Sie zum Beispiel einen Remote-Zugriff über SSH oder RDP ermöglichen wollen, können Sie NAT-Regeln definieren, um den Datenverkehr von einem externen Port an eine einzige VM weiterzuleiten.
- *Backend-IP-Pool* – mit diesem sind die VMs verbunden, auf denen Ihre Anwendung ausgeführt wird. Backend-Pools sind Lastenausgleichsregeln zugeordnet. Sie können verschiedene Backend-Pools für verschiedene Teile Ihrer Anwendungen erstellen.

Azure Application Gateway: Erweiterter Lastausgleich

Azure Load Balancer können auf der Netzwerk- oder der Anwendungsschicht arbeiten. In diesem Kapitel geht es vor allem um den regulären Azure Load Balancer, der auf der Netzwerkschicht arbeitet (Schicht 4 oder Transportprotokoll). In dieser Schicht wird der Datenverkehr untersucht und verteilt, der Load Balancer kennt aber weder die Bedeutung des Datenverkehrs noch die von Ihnen ausgeführten Anwendungen.

Azure Application Gateway ist ein Load Balancer, der auf der Anwendungsschicht (Schicht 7) arbeitet. Application Gateway hat Einblick in die Anwendung, die auf der VM ausgeführt wird, und kann die Datenverkehrsströme genauer steuern. Ein großer Vorteil von Application Gateway besteht in der Möglichkeit, verschlüsselten HTTPS-Webdatenverkehr zu behandeln.

Wenn Sie einen Lastenausgleich für Websites mit SSL-Zertifikaten durchführen, können Sie den Prozess auslagern, der den von den Webservern stammenden Datenverkehr überprüft und entschlüsselt. Auf Websites mit viel SSL-Datenverkehr kann der Prozess der Überprüfung und Entschlüsselung des Datenverkehrs einen großen Teil der Rechenzeit der VMs oder Web Apps verbrauchen. Application Gateway kann den Datenverkehr überprüfen und entschlüsseln, die reine Webanfrage an die Webserver übergeben und den von den Webservern empfangenen Datenverkehr dann wieder verschlüsseln und an Kund*innen zurückgeben.

Application Gateway bietet einige weitere, fortschrittliche Lastenausgleichsfunktionen: Die Möglichkeit, den Datenverkehr an einen beliebigen IP-Endpunkt zu verteilen, und nicht nur an eine Azure-VM. Wenn Sie Anwendungen entwickeln, die nicht nur VMs verwenden, können sich diese erweiterten Verteilungsregeln als nützlich erweisen. Dabei gelten dieselben Kernkonzepte wie bei einem regulären Lastausgleich. Auf diesen konzentrieren wir uns in diesem Kapitel, damit Sie das Funktionsprinzip in Azure kennen.

8.1.1 Erstellen eines Frontend-IP-Pools

In früheren Kapiteln haben Sie VMs erstellt, die eine direkt zugewiesene öffentliche IP-Adresse hatten. Sie haben diese öffentliche IP-Adresse verwendet, um über eine Remote-Verbindung, z. B. SSH oder RDP, auf die VM zuzugreifen. Oder Sie haben einen Webbrowser benutzt, um auf eine Website zuzugreifen, die auf der VM ausgeführt wurde. Wenn Sie einen Load Balancer nutzen, stellen Sie keine direkte Verbindung zu den VMs her. Damit der Datenverkehr stattdessen Ihren Load Balancer erreichen und an Ihre VMs verteilt werden kann, müssen der externen Schnittstelle des Load Balancers eine oder mehrere IP-Adressen zugeordnet werden.

Load Balancer können in einem von zwei verschiedenen Modi arbeiten:

- *Internetlastausgleich* – eine oder mehrere *öffentliche* IP-Adressen sind mit dem Frontend-IP-Pool verbunden. Ein Internet Load Balancer erhält den Datenverkehr direkt aus dem Internet und verteilt ihn an Backend-VMs. Ein gängiges Beispiel sind Frontend-Webserver, auf die Kund*innen direkt über das Internet zugreifen.
- *Interner Lastausgleich* – eine oder mehrere *private* IP-Adressen sind mit dem Frontend-IP-Pool verbunden. Ein interner Load Balancer arbeitet in einem virtuellen Azure-Netzwerk, zum Beispiel für Backend-Datenbank-VMs. In der Regel sind Backend-Datenbanken oder -Anwendungen nicht für die Außenwelt sichtbar. Stattdessen verbindet sich eine Reihe von Front-End-Webservern mit einem internen Load Balancer, der den Datenverkehr ohne direkten öffentlichen Zugang verteilt. Abbildung 8.2 zeigt, wie ein interner Load Balancer den Datenverkehr an Backend-VMs verteilt, die sich hinter einem auf der öffentlichen Seite arbeitenden Load Balancer und Frontend-Web-VMs befinden.

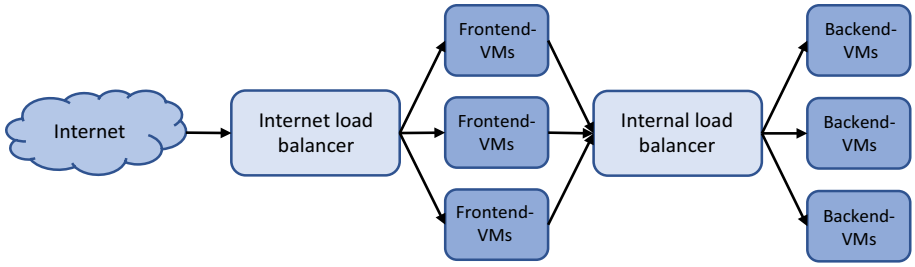


Abbildung 8.2 Mit einem Internet Load Balancer kann Datenverkehr an Frontend-VMs verteilt werden, auf denen Ihre Website ausgeführt wird, die sich dann mit einem internen Load Balancer verbinden, um den Datenverkehr an VMs auf der Datenbankebene zu verteilen. Der interne Load Balancer ist nicht öffentlich zugänglich und kann nur von den Frontend-VMs innerhalb des virtuellen Azure-Netzwerks erreicht werden.

Der Modus Ihres Load Balancers verändert das Verhalten des Frontend-IP-Pools nicht. Sie weisen eine oder mehrere IP-Adressen zu, die verwendet werden, sobald der Zugriff auf den Load Balancer angefordert wird. Für den Frontend-IP-Pool können sowohl IPv4- als auch IPv6-Adressen konfiguriert werden. Damit können Sie den eingehenden und ausgehenden Datenverkehr des Load Balancers für eine durchgängige IPv6-Kommunikation zwischen Kund*innen und Ihren VMs konfigurieren.

Jetzt testen

Sehen wir uns an, wie die Load Balancer-Komponenten zusammenarbeiten. Führen Sie hierzu die folgenden Schritte aus, um einen Load Balancer und einen Frontend-IP-Pool zu erstellen:

1. Öffnen Sie das Azure Portal, und klicken Sie oben im Dashboard auf das Cloud Shell-Symbol.
2. Erstellen Sie mit dem Befehl `az group create` eine Ressourcengruppe.

   Geben Sie einen Ressourcengruppennamen, z. B. `azuremolchapter8`, und einen Standort ein:

   ```
   az group create --name azuremolchapter8 --location westeurope
   ```

   Da wir auch hier auf Kapitel 7 aufbauen und Verfügbarkeitszonen nutzen möchten, achten Sie auf die Region, die Sie auswählen, um sicherzustellen, dass diese Verfügbarkeitszonen unterstützt werden.
3. Erstellen Sie mit dem Befehl `az network public-ip create` eine öffentliche IP-Adresse.

   In Kapitel 7 haben Sie gelernt, dass Verfügbarkeitszonen für die Redundanz von Netzwerkressourcen sorgen. Erstellen Sie daher eine zonenredundante öffentliche IP-Adresse, und geben Sie einen Namen an, z. B. `publicip`.

   ```
   az network public-ip create \
     --resource-group azuremolchapter8 \
     --name publicip \
     --sku standard
   ```

Um eine öffentliche IPv6-Adresse zu erstellen, können Sie zum vorherigen Befehl `--version IPv6` hinzufügen. Für diese Übungen können Sie IPv4-Adressen verwenden.

4 Erstellen Sie den Load Balancer, und weisen Sie die öffentliche IP-Adresse dem Frontend-IP-Pool zu. Geben Sie den Parameter `--public-ip-address` an, um die öffentliche IP-Adresse hinzuzufügen. Um einen internen Load Balancer zu erstellen, müssten Sie stattdessen den Parameter `--private-ip-address` angeben.

Erstellen Sie, wie für die öffentliche IP-Adresse, einen zonenredundanten Load Balancer vom Typ „standard", der über die Verfügbarkeitszonen hinweg arbeitet:

```
az network lb create \
  --resource-group azuremolchapter8 \
  --name loadbalancer \
  --public-ip-address publicip \
  --frontend-ip-name frontendpool \
  --backend-pool-name backendpool \
  --sku standard
```

Auf den nächsten Seiten sehen wir uns den Backend-Pool im Detail an.

8.1.2 Erstellen und Konfigurieren von Integritätstests

Wenn in einer der VMs, auf denen Ihre Anwendung ausgeführt wird, ein Problem auftritt, sollte der Load Balancer den Datenverkehr dann weiterhin an diese VM verteilen? Ein Kunde, der versucht, auf Ihren Pizzaladen zuzugreifen, könnte an diese VM weitergeleitet werden und könnte dort dann kein Essen bestellen! Ein Load Balancer überwacht den Status der VMs und kann VMs entfernen, wenn dort Probleme auftreten. Der Load Balancer überwacht den Zustand auch weiterhin und fügt die VM wieder in den Pool für die Verteilung des Datenverkehrs ein, sobald diese wieder ordnungsgemäß reagiert.

Ein Integritätstest kann in verschiedenen Modi ausgeführt werden:

- *Auf Basis des Ports* – der Load Balancer prüft, ob eine VM an einem bestimmten Port mit einem bestimmten Protokoll antwortet, z. B. an TCP-Port 80. Solange die VM auf den Integritätstest an TCP-Port 80 reagiert, wird Datenverkehr vom Load Balancer an die VM verteilt. Ansonsten wird die VM aus der Verteilung des Datenverkehrs entfernt, wie in Abbildung 8.3 dargestellt. Dieser Modus bietet keine Garantie dafür, dass die VM den Datenverkehr erwartungsgemäß bedient, sondern nur dafür, dass die Netzwerkverbindung besteht und der Zieldienst eine Antwort zurückgibt.
- *Auf Basis des HTTP-Pfads* – auf jeder VM wird eine benutzungsdefinierte Seite platziert, z. B. „health.html". Dieser benutzungsdefinierte Integritätstest kann verwendet werden, um den Zugriff auf einen Bildspeicher oder eine Datenbankverbindung zu prüfen. In diesem Modus verbleibt die VM nur dann in der Lastenausgleichsverteilung des Datenverkehrs, wenn die Seite den HTTP-Code 200 als Antwort zurückgibt, wie in Abbildung 8.4 dargestellt. Bei einem portbasierten Integritätstest ist es möglich, dass der eigentliche Webserver läuft, aber keine Datenbankverbindung hat. Mit einer benutzungsdefinierten Seite für den Integritätstest kann der Load Balancer bestätigen, dass die VM in der Lage ist, dem Kunden echten Datenverkehr zu liefern.

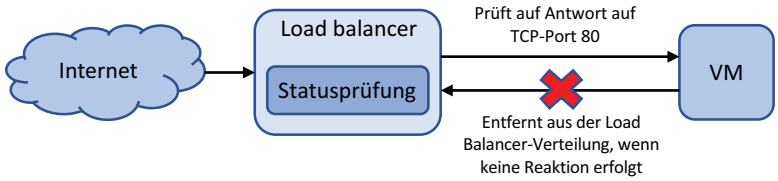


Abbildung 8.3 Ein auf dem Port basierender Load Balancer-Integritätstest prüft, ob am definierten Port mit dem vorgegebenen Protokoll eine Antwort der VM erfolgt. Wenn die VM nicht im vorgegebenen Zeitraum antwortet, wird sie aus der Lastausgleichsverteilung des Datenverkehrs entfernt. Sobald die VM wieder ordnungsgemäß reagiert, erkennt der Integritätstest die Änderung und fügt die VM wieder zur Load Balancer-Datenverkehrsverteilung hinzu.

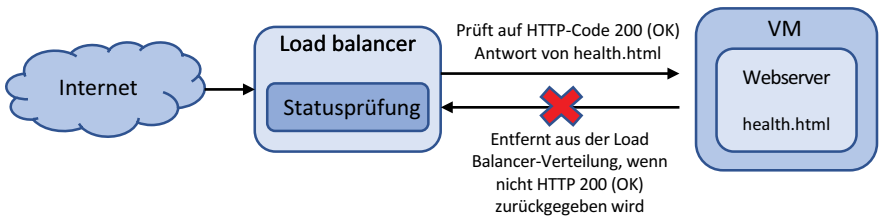


Abbildung 8.4 Eine VM, auf der ein Webserver ausgeführt wird und die eine benutzerdefinierte Seite „health.html" besitzt, bleibt so lange in der Lastausgleichsverteilung des Datenverkehrs, wie der Integritätstest den HTTP-Code 200 (OK) als Antwort erhält. Wenn im Webserverprozess ein Problem auftritt und angeforderte Seiten nicht zurückgegeben werden können, werden diese Seiten aus der Lastenausgleichsverteilung des Datenverkehrs entfernt. Dieser Prozess bietet eine gründlichere Prüfung des Zustands des Webservers als portbasierte Integritätstests.

Das Erstellen der benutzungsdefinierten Seite für den Integritätstest erfordert zwar zusätzliche Arbeit, die verbesserte Customer Experience lohnt aber den Aufwand. Die Seite für den Integritätstest muss nicht kompliziert sein. Es kann sich um eine ganz einfache HTML-Seite handeln, die verwendet wird, um zu bestätigen, dass der Webserver Seiten bedienen kann. Ohne die Seite für den Integritätstest wäre die VM weiterhin an TCP-Port 80 verfügbar, auch wenn mit dem Webserverprozess ein Problem vorliegt. Der portbasierte Integritätstest würde daher annehmen, dass die VM ordnungsgemäß läuft. Bei einem auf dem HTTP-Pfad basierenden Integritätstest muss der Webserver eine korrekte HTTP-Antwort zurückgeben. Wenn der Webserverprozess abstürzt oder ein Fehler auftritt, wird keine HTTP-Antwort gesendet und die VM wird aus der Lastausgleichsverteilung des Datenverkehrs entfernt.

Über zwei Parameter können Sie auch konfigurieren, wie oft der Integritätstest die VM überprüft und wie die Antwort aussehen muss:

- *Intervall* – legt fest, wie oft der Integritätstest den Status der VM überprüft. Standardmäßig prüft der Integritätstest den Status alle 15 Sekunden.
- *Grenzwert* – dieser legt fest, wie viele aufeinanderfolgende Antwortfehler der Integritätstest erhalten muss, bis die VM aus der Lastausgleichsverteilung des Datenverkehrs entfernt wird. Standardmäßig toleriert der Integritätstest zwei aufeinanderfolgende Fehler, bevor die VM aus der Lastenausgleichsverteilung des Datenverkehrs entfernt wird.

Jetzt testen

Führen Sie die folgenden Schritte aus, um einen Integritätstest für Ihren Lastenausgleich zu erstellen, wie in Abbildung 8.4 dargestellt:

1. Öffnen Sie das Azure Portal, und klicken Sie oben im Dashboard auf das Cloud Shell-Symbol.
2. Geben Sie einen Namen für den Integritätstest an, z. B. `healthprobe`. Um den Integritätstest für einen Webserver einzurichten, geben Sie den HTTP-Port 80 an, und definieren Sie eine eigene Integritätstestseite namens „health.html". In Abschnitt 8.2 erstellen Sie diese Integritätstestseite auf Ihren VMs Definieren Sie ein Intervall von 10 Sekunden und einen Grenzwert von drei aufeinanderfolgenden Fehlern, um die Konfiguration dieser Werte zu testen:

```
az network lb probe create \
  --resource-group azuremolchapter8 \
  --lb-name loadbalancer \
  --name healthprobe \
  --protocol http \
  --port 80 \
  --path health.html \
  --interval 10 \
  --threshold 3
```

Wie können Sie jetzt, nachdem der Integritätstest erstellt wurde, den Status Ihrer VMs überprüfen? Integritätstests werden Lastausgleichsregeln zugeordnet. Derselbe Integritätstest kann mit verschiedenen Lastenausgleichsregeln verwendet werden. Erinnern Sie sich an Kapitel 5, in dem Sie Netzwerksicherheitsgruppen (NSGs) und -regeln erstellt haben? Diese NSGs können mehreren VMs oder virtuellen Netzwerk-Subnetzen zugeordnet werden. Eine ähnliche Eins-zu-viele-Beziehung gilt auch für Integritätstests.

Sehen wir uns an, wie wir unseren Integritätstest zum Laufen bringen und Lastenausgleichsregeln erstellen.

8.1.3 Definieren der Datenverkehrsverteilung mit Load Balancer-Regeln

Wenn der Datenverkehr durch den Load Balancer an Backend-VMs weitergeleitet wird, können Sie festlegen, unter welchen Bedingungen auch der oder Benutzer*in an dieselbe VM weitergeleitet wird. Vielleicht möchten Sie, dass Benutzer*innen für die Dauer einer einzigen Sitzung eine Verbindung zu derselben VM behalten. Oder Sie möchten Benutzer*innen erlauben, zurückzukehren und ihre VM-Sitzungsaffinität auf der Grundlage der Quell-IP-Adresse beizubehalten. Abbildung 8.5 zeigt ein Beispiel für den standardmäßigen Session-Affinity-Modus.

Im Session-Affinity-Modus wird der Datenverkehrsstrom mit einem 5-Tuple-Hash abgewickelt, der die Quell-IP-Adresse, den Quellport, die Ziel-IP-Adresse, den Zielport und den Protokolltyp verwendet. Im Grunde genommen wird ein Benutzender für jede Anfrage, die er über TCP-Port 80 an Ihren Webserver stellt, für die Dauer der Sitzung an dieselbe Backend-VM weitergeleitet.

Was passiert, wenn der Kunde seine Browser-Sitzung schließt? Wenn er das nächste Mal eine Verbindung herstellt, wird eine neue Sitzung gestartet. Da der Load Balancer den Datenverkehr über alle ordnungsgemäß funktionierenden VMs im Backend-IP-Pool verteilt, ist es möglich, dass der oder Benutzer*in wieder mit derselben VM verbunden wird. Je mehr VMs Sie aber im Backend-IP-Pool haben, desto größer ist die Chance, dass der oder die Benutzer*in mit einer anderen VM verbunden wird.

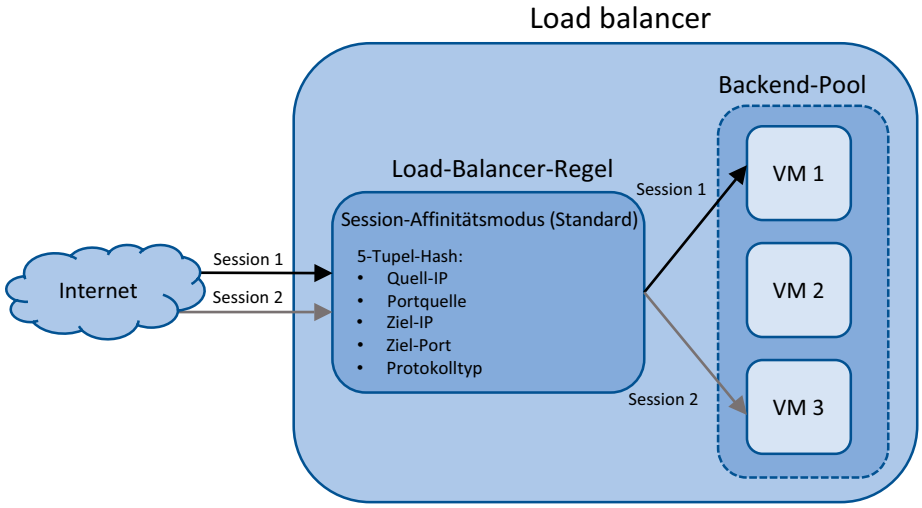


Abbildung 8.5 Im Session-Affinity-Modus verbindet sich ein Benutzender nur für die Dauer seiner Session mit derselben Backend-VM.

Für Anwendungsbesitzer*innen und -entwickler*innen kann es wünschenswert sein, dass Benutzer*innen mit derselben VM verbunden werden, wenn sie eine weitere Sitzung starten. Wenn Ihre Anwendung z. B. Dateiübertragungen abwickelt oder UDP anstelle von TCP verwendet, dann möchten Sie wahrscheinlich, dass weiterhin dieselbe VM ihre Anfragen verarbeitet. In diesen Szenarien können Sie die Load Balancer-Regeln für die Quell-IP-Affinität konfigurieren. Abbildung 8.6 zeigt ein Beispiel für den Quell-IP-Affinitätsmodus.

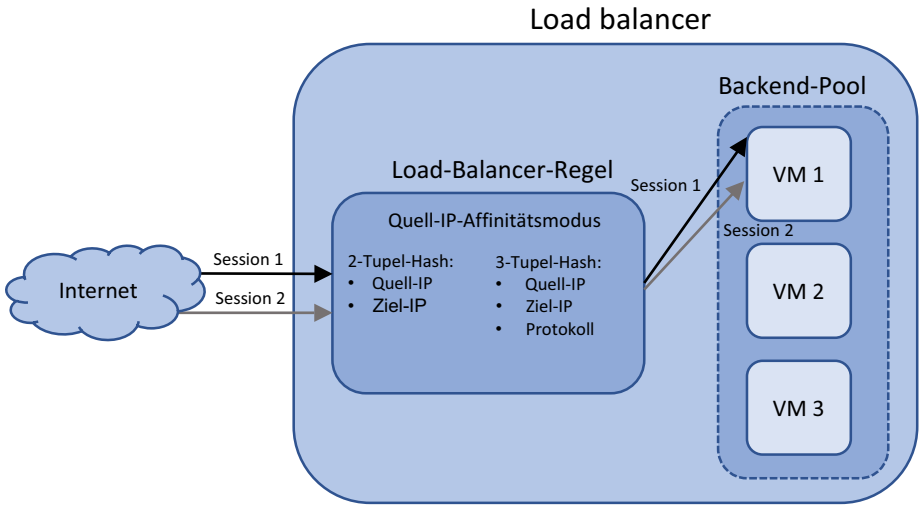


Abbildung 8.6 Wenn Sie die Load Balancer-Regeln so konfigurieren, dass der Quell-IP-Affinitätsmodus verwendet wird, kann der oder die Benutzer*in eine Sitzung schließen und dann eine neue Sitzung starten, die ihn wieder mit derselben Back-End-VM verbindet. Der Quell-IP-Affinitätsmodus kann einen 2-Tuple-Hash verwenden, der die Quell- und Ziel-IP-Adresse verwendet, oder einen 3-Tuple-Hash, der außerdem das Protokoll verwendet.

Jetzt testen

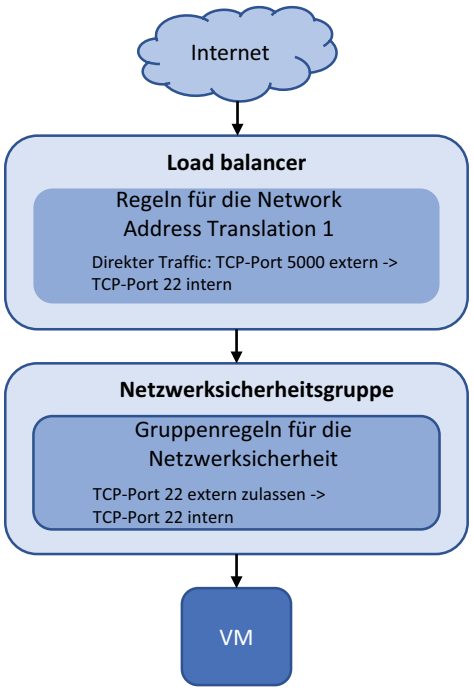
Führen Sie die folgenden Schritte aus, um eine Load Balancer-Regel zu erstellen, die einen Integritätstest verwendet.

1. Öffnen Sie das Azure Portal, und klicken Sie oben im Dashboard auf das Cloud Shell-Symbol.
2. Um eine Load Balancer-Regel zu erstellen, geben Sie einen Namen für die Regel an, z. B. `httprule`.
3. Geben Sie den externen Port an, an dem der Datenverkehr empfangen wird, und den internen Port, an den der Datenverkehr verteilt wird. In diesem einfachen Beispiel wird der Datenverkehr an Port 80 empfangen und dann an Port 80 verteilt:

```
az network lb rule create \
  --resource-group azuremolchapter8 \
  --lb-name loadbalancer \
  --name httprule \
  --protocol tcp \
  --frontend-port 80 \
  --backend-port 80 \
  --frontend-ip-name frontendpool \
  --backend-pool-name backendpool \
  --probe-name healthprobe
```

Wenn Sie mehrere Websites auf einer VM ausführen, die an verschiedenen Ports antwortet, kann eine Regel den Datenverkehr direkt an eine bestimmte Website auf der VM weiterleiten.

8.1.4 Weiterleitung von direktem Datenverkehr mit NAT-Regeln



Die Load Balancer-Regeln verteilen den Datenverkehr über die Backend-Pools der VMs, sodass es keine Garantie dafür gibt, dass Sie zu Wartungs- oder Verwaltungszwecken eine Verbindung zu einer bestimmten VM herstellen können. Wie lässt sich eine Verbindung zu einer bestimmten VM herstellen, die sich hinter einem Load Balancer befindet? Ein letzter Teil der Load Balancer-Konfiguration, den wir behandeln, betrifft Netzwerkadressübersetzungsregeln (NAT-Regeln), mit denen Sie einen spezifischen Datenverkehrsfluss direkt an eine einzige VM weiterleiten können. Abbildung 8.7 zeigt, wie NAT-Regeln einen spezifischen Datenverkehr an einzelne VMs weiterleiten.

Abbildung 8.7 Der Datenverkehr wird im Load Balancer über NAT-Regeln verarbeitet. Wenn ein Protokoll und ein Port mit einer Regel übereinstimmen, wird der Datenverkehr an die definierte Backend-VM weitergeleitet. Es sind keine Integritätstests festgelegt, sodass der Load Balancer nicht prüft, ob die VM reagiert, bevor er den Datenverkehr weiterleitet. Der Datenverkehr verlässt den Load Balancer und wird dann über NSG-Regeln verarbeitet. Sofern der Datenverkehr zulässig ist, wird er an die VM weitergeleitet.

NAT-Regeln funktionieren neben NSG-Regeln. Die VM kann den Datenverkehr nur erhalten, wenn es eine NSG-Regel gibt, die denselben Datenverkehr zulässt wie die NAT-Regel für den Lastausgleich.

Warum sollten Sie NAT-Regeln erstellen? Was wäre, wenn Sie für die Verbindung mit einer bestimmten VM SSH oder RDP verwenden wollten (und nicht Azure Bastion, von dem ich in Kapitel 2 gesprochen habe)? Oder wenn Sie Verwaltungstools verwenden wollten, um sich mit einem Backend-Datenbankserver zu verbinden? Wenn der Load Balancer den Datenverkehr über die Backend-VMs hinweg verteilt, müssten Sie immer wieder versuchen, sich zu verbinden, und wären dann möglicherweise immer noch nicht mit der gewünschten VM verbunden.

Sicherheitsaspekte

Im dritten Teil des Buchs werden wir uns mit einigen Sicherheitsthemen befassen. Die Sicherheit sollte aber ein allgegenwärtiges Thema sein, wenn Sie Anwendungen in Azure erstellen und ausführen. Bei der Sicherheit sollte es sich nicht um etwas handeln, das später hinzufügt wird. Mit dem Aufkommen von Cloud Computing und verfügbaren VMs und Web Apps können einige grundlegende Best Practices für die Sicherheit leicht übersehen werden. Insbesondere wenn Sie im Rahmen eines größeren Unternehmensabonnements in Azure arbeiten, stellen Sie sicher, dass keine der von Ihnen erstellten Ressourcen Angreifern unbeabsichtigt eine Möglichkeit bietet, Zugang zu Ihrer Infrastruktur zu erhalten.

Was sollte vermieden werden? Nun, einige der Dinge, die Sie in diesem Buch bereits gemacht haben! Remote-Verwaltungsports für SSH und RDP sollten nicht, wie wir es gemacht haben, für das öffentliche Internet geöffnet werden. Oder zumindest sollte der Zugriff auf einen bestimmten IP-Adressbereich beschränkt werden.

Am besten verwenden Sie einen verwalteten Dienst wie Azure Bastion oder erstellen manuell eine gesicherte VM, die über eine Remote-Verwaltung verfügt. Im Bedarfsfall verbinden Sie sich mit dem Azure Bastion-Host oder Ihrer gesicherten VM und stellen dann über das interne virtuelle Azure-Netzwerk eine Verbindung zu weiteren VMs her. In Kapitel 5 haben Sie den einfachen Jumpbox-Ansatz verwendet. Dieser Ansatz minimiert die Angriffsfläche und verringert die Notwendigkeit von NSG-Regeln und NAT-Regeln für den Lastausgleich. In Kapitel 16 werden wir über Azure Security Center sprechen und wie Sie Remoteverwaltungsports dynamisch für einen bestimmten Zeitraum anfordern und öffnen können. Dieser Ansatz vereint das Beste aus beiden Welten.

Versuchen Sie, auch wenn Sie in einem privaten Azure-Abonnement arbeiten, das keine Konnektivität zu anderen Azure-Abonnements in der Schule oder am Arbeitsplatz besitzt, die bereitgestellte Remote-Konnektivität möglichst zu minimieren.

Jetzt testen

Führen Sie die folgenden Schritte aus, um eine NAT-Regel für den Load Balancer zu erstellen:

1. Öffnen Sie das Azure Portal, und klicken Sie oben im Dashboard auf das Cloud Shell-Symbol.
2. Legen Sie für eine NAT-Regel für den Load Balancer einen Namen fest, z. B. `natrulessh`, und den zu verwendenden Frontend-IP-Pool. Die NAT-Regel untersucht den Datenverkehr für ein bestimmtes Protokoll und einen bestimmten Port, z. B. TCP-Port 50001. Wenn eine Übereinstimmung mit der Regel vorliegt, wird der Datenverkehr an Back-End-Port 22 weitergeleitet:

```
az network lb inbound-nat-rule create \
  --resource-group azuremolchapter8 \
  --lb-name loadbalancer \
  --name natrulessh \
  --protocol tcp \
  --frontend-port 50001 \
  --backend-port 22 \
  --frontend-ip-name frontendpool
```

Damit haben Sie einen einfachen Load Balancer erstellt. Sehen Sie sich die Load Balancer-Komponenten an:

```
az network lb show \
  --resource-group azuremolchapter8 \
  --name loadbalancer
```

Dem Frontend-IP-Pool wurde eine öffentliche IP-Adresse zugewiesen. Und Sie haben einen Integritätstest erstellt, der den Status auf einer benutzerdefinierten Zustandsseite für einen Webserver überprüft. Eine Load Balancer-Regel wurde erstellt, um den Webdatenverkehr von Ihren Kund*innen an einen Backend-Pool zu verteilen und den Integritätstest anzuwenden. Sie haben außerdem eine Load Balancer-NAT-Regel, die SSH-Datenverkehr erlaubt. Es gibt aber noch keine VMs, die diesen Datenverkehr erhalten. Ihre Pizzakund*innen sind hungrig. Lassen Sie uns daher einige VMs erstellen, auf denen Ihre Webanwendung ausgeführt wird und an die der Load Balancer den Datenverkehr verteilen kann!

8.1.5 Zuweisen von Gruppen von VMs zu Backend-Pools

Im letzten Abschnitt des Lastenausgleichs werden Backend-Pools definiert, die eine oder mehrere VMs enthalten. Diese Backend-Pools enthalten VMs, auf denen die gleichen Anwendungskomponenten ausführt werden. So kann der Load Balancer den Datenverkehr an einen bestimmten Backend-Pool verteilen und darauf vertrauen, dass alle VMs in diesem Pool ordnungsgemäß auf die Kundenanfrage reagieren können. In Abbildung 8.8 wird detailliert dargestellt, wie die Backend-Pools die VMs logisch gruppieren, auf denen die gleichen Anwendungen ausgeführt werden.

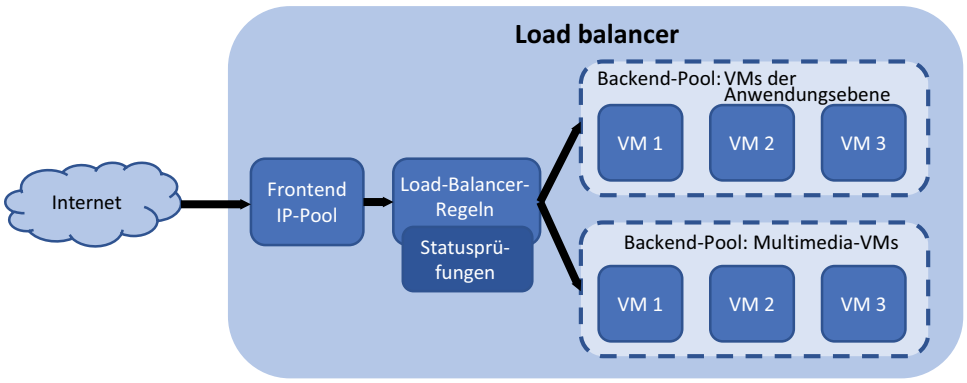


Abbildung 8.8 Sie können in einem Load Balancer einen oder mehrere Backend-Pools erstellen. Jeder Backend-Pool enthält eine oder mehrere VMs, auf denen die gleiche Anwendungskomponente ausgeführt wird. In diesem Beispiel enthält ein Backend-Pool die VMs, auf denen die Webanwendung ausgeführt wird. Ein weiterer Backend-Pool enthält die VMs, die Multimediaelemente, wie Bilder und Videos, bereitstellen.

Sie erstellen und verwenden zwar einen Load Balancer mit VMs, alles läuft aber auf der virtuellen Netzwerkebene ab. Der Frontend-IP-Pool verwendet öffentliche oder private IP-Adressen. Der Integritätstest prüft die Antworten eines bestimmten Ports oder Protokolls. Auch wenn ein HTTP-Test angewendet wird, sucht der Load Balancer nach einer positiven Netzwerkantwort. Lastausgleichsregeln konzentrieren sich darauf, wie der Datenverkehr von einem externen Port im Frontend-Pool an einen Port im Backend-Pool verteilt wird.

Wenn Sie VMs dem Backend-Pool zuweisen, die den vom Load Balancer verteilten Datenverkehr erhalten, ist es die virtuelle NIC, die sich mit dem Load Balancer verbindet. Die VM wird zufällig mit der virtuellen NIC verbunden. Denken Sie an Kapitel 5 zurück. Diese Trennung von VMs und virtueller NIC ist sinnvoll im Hinblick darauf, wie die Ressourcen verwaltet werden. NSG-Regeln kontrollieren, welcher Datenverkehr in die VM gelangt, sie werden aber auf ein virtuelles Subnetz des Netzwerks oder virtuelle NIC angewendet, nicht auf die VM.

Was bedeutet das für die Konfiguration von Backend-IP-Pools? Sie müssen Ihre restlichen virtuellen Netzwerkressourcen erstellen, bevor Sie eine VM mit dem Load Balancer verbinden können. Die Schritte zum Erstellen der Netzwerkressourcen fassen zusammen, was Sie in früheren Kapiteln gelernt haben. Schauen wir mal, woran Sie sich noch erinnern!

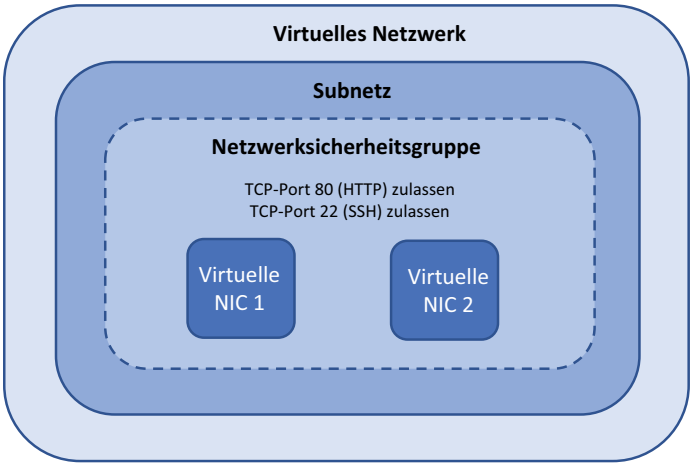


Abbildung 8.9 Um das virtuelle Netzwerk vorzubereiten, erstellen Sie in dieser Übung ein Netzwerk, ein Subnetz und virtuelle NICs, die durch eine NSG geschützt sind. Mit der NSG verbundene Regeln erlauben den HTTP- und SSH-Datenverkehr.

Jetzt testen

Führen Sie die folgenden Schritte aus, um die zusätzlichen Netzwerkressourcen zu erstellen, wie in Abbildung 8.9 dargestellt:

1. Öffnen Sie das Azure Portal, und klicken Sie oben im Dashboard auf das Cloud Shell-Symbol.
2. Erstellen Sie ein virtuelles Netzwerk und ein Subnetz:

```
az network vnet create \
  --resource-group azuremolchapter8 \
```

```
  --name vnetmol \
  --address-prefixes 10.0.0.0/16 \
  --subnet-name subnetmol \
  --subnet-prefix 10.0.1.0/24
```

In der Praxis ist es sehr gut möglich, dass diese Netzwerkressourcen bereits vorhanden sind. Es handelt sich auch um dieselben Namen und IP-Adressbereiche, die Sie in Kapitel 5 verwendet haben. Da Sie am Ende jedes Kapitels die Azure-Ressourcen bereinigt haben, sollte die Wiederverwendung der IP-Bereiche kein Problem darstellen. Denken Sie nur daran, dass Sie normalerweise kein virtuelles Netzwerk und Subnetz erstellen, wenn Sie einen Load Balancer erstellen. Sie können vielmehr die bereits vorhandenen virtuellen Netzwerkressourcen nutzen.

3 Erstellen Sie eine NSG:

```
az network nsg create \
  --resource-group azuremolchapter8 \
  --name webnsg
```

4 Erstellen Sie eine NSG-Regel, die zulässt, dass der von TCP-Port 80 stammende Datenverkehr zu Ihren VMs gelangt. Diese Regel wird für die Webserver-VMs benötigt, damit sie den Kundendatenverkehr empfangen und beantworten können:

```
az network nsg rule create \
  --resource-group azuremolchapter8 \
  --nsg-name webnsg \
  --name allowhttp \
  --priority 100 \
  --protocol tcp \
  --destination-port-range 80 \
  --access allow
```

5 Fügen Sie eine weitere Regel hinzu, um den SSH-Datenverkehr für die Remoteverwaltung zuzulassen. Diese NSG-Regel arbeitet mit der Load Balancer-NAT-Regel zusammen, die Sie in Abschnitt 8.1.4 für eine Ihrer VMs erstellt haben:

```
az network nsg rule create \
  --resource-group azuremolchapter8 \
  --nsg-name webnsg \
  --name allowssh \
  --priority 101 \
  --protocol tcp \
  --destination-port-range 22 \
  --access allow
```

6 Ordnen Sie die NSG dem in Schritt 2 erstellten Subnetz zu. NSG-Regeln werden auf alle VMs angewendet, die sich mit diesem Subnetz verbinden:

```
az network vnet subnet update \
  --resource-group azuremolchapter8 \
  --vnet-name vnetmol \
  --name subnetmol \
  --network-security-group webnsg
```

7 Der Load Balancer arbeitet mit virtuellen NICs zusammen. Erstellen Sie daher zwei virtuelle NICs, und verbinden Sie diese mit dem Subnetz des virtuellen

Netzwerks. Geben Sie außerdem den Namen des Load Balancers und den Backend-Adresspool an, mit dem sich die virtuellen NICs verbinden. Die NAT-Regel des Load Balancers wird nur mit dieser ersten virtuellen NIC verbunden, die Sie erstellen:

```
az network nic create \
  --resource-group azuremolchapter8 \
  --name webnic1 \
  --vnet-name vnetmol \
  --subnet subnetmol \
  --lb-name loadbalancer \
  --lb-address-pools backendpool \
  --lb-inbound-nat-rules natrulessh
```

8 Erstellen Sie die zweite NIC auf die gleiche Weise, aber ohne die NAT-Regel des Load Balancers:

```
az network nic create \
  --resource-group azuremolchapter8 \
  --name webnic2 \
  --vnet-name vnetmol \
  --subnet subnetmol \
  --lb-name loadbalancer \
  --lb-address-pools backendpool
```

8.2 Erstellen und Konfigurieren von VMs mit dem Load Balancer

Machen wir eine Pause und sehen uns an, was wir bis jetzt erstellt haben. Abbildung 8.10 zeigt eine Gesamtübersicht unserer Netzwerkressourcen und des Load Balancers. Beachten Sie, wie sehr diese Ressourcen integriert sind. Der Load Balancer kann nicht alleine existieren. Virtuelle NICs müssen mit dem Load Balancer verbunden sein,

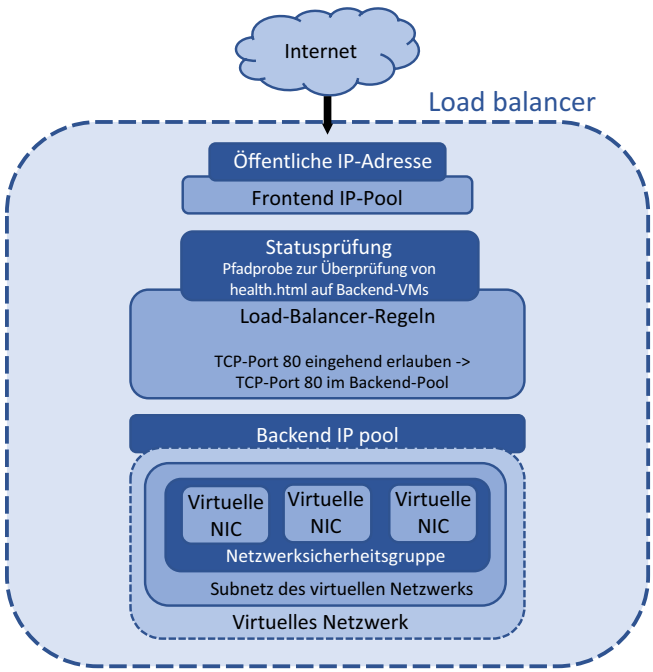


Abbildung 8.10 Hier wurden keine VMs erstellt – bei der Load-Balancer-Konfiguration geht es um virtuelle Netzwerkressourcen. Es besteht eine enge Beziehung zwischen dem Load Balancer und virtuellen Netzwerkressourcen.

damit Datenverkehr verteilt werden kann. Diese virtuellen NICs benötigen ein virtuelles Netzwerk und ein Subnetz und werden idealerweise durch eine NSG geschützt. Die VMs, auf denen Ihre Anwendung läuft, haben mit den Schritten zum Erstellen und Konfigurieren des Load Balancers fast nichts zu tun!

Sie haben eine Menge Netzwerkressourcen erstellt und mehrere Bereiche des Load Balancers konfiguriert. Die öffentliche IP-Adresse und der Load Balancer wurden als zonenredundante Ressourcen in einer Verfügbarkeitszone erstellt. Wir können also zwei VMs in verschiedenen Zonen erstellen, um die Hochverfügbarkeit unserer Anwendungen durch Verfügbarkeitszonen weiter zu verstärken.

Wenn Sie anstelle von Verfügbarkeitszonen Verfügbarkeitsgruppen verwenden, müssen Sie zuerst eine Verfügbarkeitsgruppe erstellen und dann die VMs hinzufügen. Die Azure-Plattform verteilt die VMs dann auf die Fehler- und Update-Domänen. Wenn Sie die Verwendung der Hochverfügbarkeit von Azure für Ihre Pizzeria maximieren wollen, dann sollten Sie aber Verfügbarkeitszonen verwenden.

Jetzt testen

Führen Sie die folgenden Schritte aus, um die VMs zu erstellen und mit dem Load Balancer zu verbinden:

1. Erstellen Sie die erste VM, und weisen Sie sie mit `--zone 1` einer Verfügbarkeitszone zu:

```
az vm create \
  --resource-group azuremolchapter8 \
  --name webvm1 \
  --image ubuntults \
  --size Standard_B1ms \
  --admin-username azuremol \
  --generate-ssh-keys \
  --zone 1 \
  --nics webnic1
```

2. Erstellen Sie die zweite VM und weisen Sie sie Verfügbarkeitszone 2 zu. Verbinden Sie die zuvor erstellte zweite virtuelle NIC mit `--nics webnic2`:

```
az vm create \
  --resource-group azuremolchapter8 \
  --name webvm2 \
  --image ubuntults \
  --size Standard_B1ms \
  --admin-username azuremol \
  --generate-ssh-keys \
  --zone 2 \
  --nics webnic2
```

Um den Load Balancer in Aktion zu erleben, müssen Sie einen einfachen Webserver installieren, wie Sie es in Kapitel 2 bereits getan haben. Sie können auch die NAT-Regel des Load Balancers ausprobieren. Sehen Sie langsam, wie all diese Komponenten in Azure miteinander zusammenhängen und aufeinander aufbauen?

Jetzt testen

In Kapitel 5 haben wir uns den SSH-Agent angesehen. Mit dem SSH-Agent können Sie einen SSH-Schlüssel von einer VM an die nächste übergeben. Nur VM1 besitzt eine NAT-Regel des Load Balancers, deshalb müssen Sie für die Verbindung mit VM2 den Agent verwenden. Führen Sie die folgenden Schritte aus, um einen Webserver auf Ihren VMs zu installieren:

1. Starten Sie den SSH-Agent, und fügen Sie Ihren SSH-Schlüssel hinzu, damit Sie mit beiden VMs eine Verbindung herstellen können:

   ```
   eval $(ssh-agent) && ssh-add
   ```

2. Rufen Sie die öffentliche IP-Adresse ab, die mit dem Frontend-IP-Pool des Load Balancers verbunden ist. Nur so kann das Routing des Datenverkehr durch die VMs erfolgen:

   ```
   az network public-ip show \
     --resource-group azuremolchapter8 \
     --name publicip \
     --query ipAddress \
     --output tsv
   ```

3. Nun sind Sie bereit, eine SSH-Verbindung zu VM 1 herzustellen. Geben Sie die öffentliche IP-Adresse des Load Balancers (ersetzen Sie `<your-ip-address>` im folgenden Befehl) und den Port an, der in der NAT-Regel des Load Balancers verwendet wurde, z. B. 50001. Der Parameter `-A` verwendet den SSH-Agent, um Ihre SSH-Schlüssel zu übergeben:

   ```
   ssh -A azuremol@<your-ip-address> -p 50001
   ```

   In Kapitel 2 haben Sie `apt-get` verwendet, um den gesamten LAMP-Stack einschließlich Apache-Webserver zu installieren. Sehen wir uns den eigenständigen, aber leistungsstarken NGINX-Webserver an, der sich ein wenig vom Apache-Webserver unterscheidet. Auf einer Windows-VM würden Sie hier IIS installieren. Führen Sie den folgenden Befehl aus, um den NGINX-Webserver zu installieren:

   ```
   sudo apt update && sudo apt install -y nginx
   ```

4. Im GitHub-Repository, das Sie in früheren Kapiteln bereits verwendet haben, finden Sie eine einfache HTML-Webseite und eine Integritätstestseite für den Integritätstest des Load Balancers. Klonen Sie diese Beispiele auf die VM:

   ```
   git clone https://github.com/foulds/azure-mol-samples-2nd-ed.git
   ```

5. Kopieren Sie die HTML-Seite und den Integritätstest in das Webserververzeichnis:

   ```
   sudo cp azure-mol-samples-2nd-ed/08/webvm1/* /var/www/html/
   ```

6. Jetzt müssen Sie eine Verbindung mit der zweiten VM herstellen und den NGINX-Webserver und Beispielcode installieren. Erinnern Sie sich an den SSH-Agent? Sie müssen eine SSH-Verbindung zur internen, privaten IP-Adresse von VM 1 zu VM 2 herstellen:

   ```
   ssh 10.0.1.5
   ```

7 Installieren Sie den NGINX-Webserver:

```
sudo apt update && sudo apt install -y nginx
```

8 Klonen Sie die GitHub-Beispiele auf die VM:

```
git clone https://github.com/fouldsy/azure-mol-samples-2nd-ed.git
```

9 Kopieren Sie die HTML-Seite und den Integritätstest in das Webserververzeichnis:

```
sudo cp azure-mol-samples-2nd-ed/08/webvm2/* /var/www/html/
```

Öffnen Sie einen Webbrowser, und stellen Sie eine Verbindung zur öffentlichen IP-Adresse Ihres Load Balancers her. Die einfache Webseite wird geladen und zeigt, dass Ihre Pizzeria jetzt redundante VMs in Verfügbarkeitszonen besitzt, die hinter einem Load Balancer laufen, wie in Abbildung 8.11 dargestellt! Möglicherweise müssen Sie das Aktualisieren Ihres Webbrowsers erzwingen, damit Sie sehen können, dass sowohl VM 1 als auch VM 2 antwortet, wenn der Load Balancer den Datenverkehr zwischen den beiden verteilt.

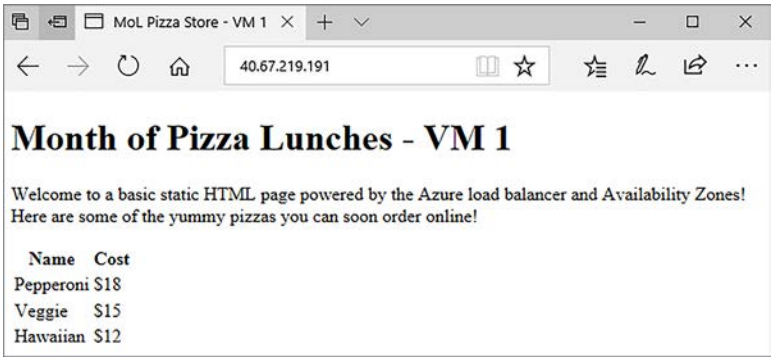


Abbildung 8.11 Wenn Sie die öffentliche IP-Adresse des Load Balancers im Webbrowser öffnen, wird der Datenverkehr an eine der VMs verteilt, auf denen Ihre einfache Website ausgeführt wird. Der Integritätstest des Load Balancers verwendet die Seite „health.html", um zu bestätigen, dass der Webserver mit dem HTTP-Code 200 (OK) antwortet. Die VM ist dann für die Verteilung des Datenverkehrs durch den Load Balancer verfügbar.

8.3 Übung: Anzeigen von Vorlagen vorhandener Bereitstellungen

Dieses Kapitel verknüpft die Themen miteinander, die Sie in den vorherigen Kapiteln kennengelernt haben. Wie in Kapitel 5 haben Sie Netzwerkressourcen erstellt. Wie in Kapitel 7 haben Sie den Load Balancer und die VMs mit Verfügbarkeitszonen hochverfügbar gemacht. Und wie in Kapitel 2 haben Sie einen Webserver installiert und Beispieldateien bereitgestellt. Ihre Pizzeria hat sich von der einfachen Webseite auf einer einzigen VM seit Beginn des Buches stark weiterentwickelt!

Um noch ein weiteres Thema aus einem früheren Kapitel zu verknüpfen, möchte ich, dass Sie in dieser Übung alle Ressourcen erkunden, die den Load Balancer ausmachen. Sehen Sie sich hierzu die Resource Manager-Vorlage an, die Sie in Kapitel 6 kennengelernt haben. Das Ziel dieser Übung besteht darin, aufzuzeigen, wie sich mit einer einzelnen Vorlage alles erstellen und konfigurieren lässt, was sonst viele Seiten

und viele CLI-Befehle erfordert. Und Sie können mir glauben: Es würde sogar noch mehr PowerShell-Befehle erfordern! Führen Sie diese Schritte aus:

1. Öffnen Sie das Azure Portal.
2. Wählen Sie auf der Navigationsleiste auf der linken Seite des Portals „Resource Group" (Ressourcengruppe) aus.
3. Wählen Sie Ihre Ressourcengruppe aus, z. B. „azuremolchapter8".
4. Wählen Sie „Export Template" (Vorlage exportieren) in der linken Navigationsleiste aus, wie in Abbildung 8.12 dargestellt.
5. Wählen Sie in der Liste die einzelnen Ressourcen aus, um den entsprechenden Teil der Vorlage anzuzeigen. Nehmen Sie sich ein paar Minuten Zeit, um diese Vorlage zu erkunden und sich alle Ressourcen und Komponenten anzusehen, die Sie in der Azure-CLI konfiguriert haben.

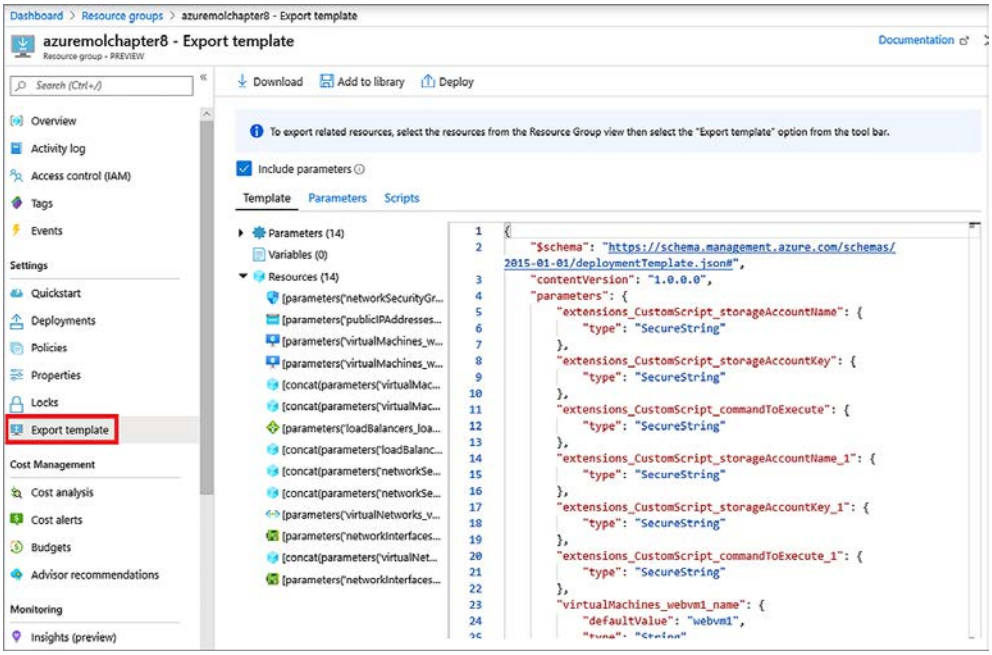


Abbildung 8.12 Wählen Sie im Azure Portal Ihre Load Balancer-Ressourcengruppe aus, und zeigen Sie die Resource Manager-Vorlage an.

Mit einer Vorlage lässt sich viel einfacher eine hochverfügbare, redundante Anwendungsumgebung mit Lastausgleich implementieren. Sie können den Namen, die Regeln und den Verteilungsmodus des Load Balancers ändern und die Vorlage bereitstellen lassen, damit sie die gesamte Anwendungsumgebung für Sie konfiguriert.

Vergessen Sie nicht, diese Ressourcengruppe zu löschen, um Ihr Azure-Guthaben zu schonen!

9 Skalierbare Anwendungen

In den beiden vorherigen Kapiteln haben wir uns angesehen, wie hochverfügbare Anwendungen entwickelt werden und wie wir mit Load Balancers Datenverkehr an mehrere VMs verteilen können, auf denen Ihre App ausgeführt wird. Wie aber können Sie mehrere VMs effizient ausführen und verwalten und die Anzahl der ausgeführten VM-Instanzen an der Nachfrage Ihrer Kund*innen ausrichten? Sobald die Kundennachfrage steigt, möchten Sie Ihre Anwendung automatisch skalieren, um damit Schritt zu halten. Und wenn die Nachfrage zurückgeht, etwa mitten in der Nacht, wenn die meisten Menschen schlafen, sofern sie keine kleinen Kinder haben, dann möchten Sie, dass die Größe der Anwendung abnimmt und damit Ihre Kosten sinken.

Auf Azure können Sie IaaS-Ressourcen mit VM-Skalierungsgruppen automatisch herunter- bzw. hochskalieren. Diese Skalierungsgruppen führen identische VMs aus, die normalerweise hinter einem Load Balancer oder einem Application Gateway verteilt sind. Sie definieren Regeln für das automatische Skalieren, über die die Anzahl der VM-Instanzen erhöht oder verringert wird, sobald sich die Kundennachfrage verändert. Load Balancer oder Application Gateways verteilen den Verkehr automatisch an die neuen VM-Instanzen. Sie können sich daher darauf konzentrieren, wie Sie Ihre Apps besser entwickeln und ausführen können. Skalierungsgruppen bieten Ihnen die Kontrolle über IaaS-Ressourcen sowie einige Vorteile von PaaS. Über Web Apps haben wir in den letzten Kapiteln noch nicht viel gesprochen. Diese treten jetzt durch ihre Fähigkeit in den Vordergrund, mit der Anwendungsnachfrage zu skalieren.

In diesem Kapitel sehen wir uns an, wie Anwendungen entwickelt und erstellt werden, die automatisch skalieren. Wir sehen uns an, warum Ihnen diese Möglichkeit des Skalierens je nach Nachfrage hilft, effiziente Anwendungen auszuführen. Außerdem lernen Sie verschiedene Verfahren, um anhand unterschiedlicher Kennzahlen zu skalieren.

9.1 Gründe für skalierbare, zuverlässige Anwendungen

Was bedeutet es, Anwendungen zu entwickeln, die skalieren? Damit können Sie mit der Kundennachfrage Schritt halten, um den steigenden Workload zu bewältigen, während Sie am Wochenende im Kino sitzen. Damit stehen Sie nicht irgendwann

mit einer Rechnung für eine Menge zusätzlicher Ressourcen da, die Sie kaum nutzen. Und damit müssen Sie auch nicht im schlimmsten Fall erleben, dass Ihre Anwendung aus Mangel an verfügbaren Ressourcen ausfällt. Der Sweet Spot für Anwendungen und die dafür benötigten Ressourcen ist selten statisch. In der Regel verändert sich die Nachfrage nach Anwendungen während der Tages- und Nachtzeiten oder an Wochentagen und Wochenenden.

Es gibt hauptsächlich zwei Verfahren, um Ressourcen zu skalieren, wie in Abbildung 9.1 dargestellt: vertikal und horizontal. Sowohl VM-Skalierungsgruppen als auch Web Apps können vertikal oder horizontal skalieren.

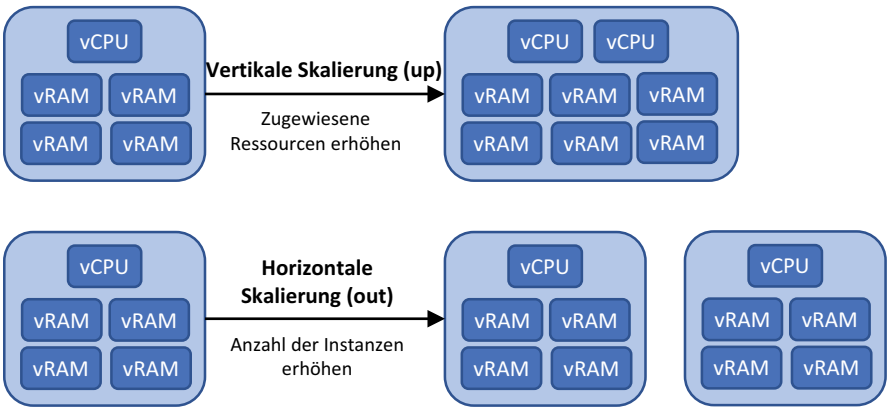


Abbildung 9.1 Sie können Ihre Anwendungen vertikal oder horizontal herunter- bzw. hochskalieren. Welche Methode Sie verwenden, hängt davon ab, wie Ihre Anwendung für den Umgang mit Skalierung ausgelegt ist. Die vertikale Skalierung passt die Ressourcen an, die einer VM oder Web App zugewiesen sind, z. B. die Anzahl der CPU-Kerne oder die Größe des Arbeitsspeichers. Diese Skalierungsmethode eignet sich gut, wenn die Anwendung nur eine Instanz ausführt. Die horizontale Skalierung verändert die Anzahl der Instanzen, die Ihre Anwendung ausführen, und verbessert so die Verfügbarkeit und die Resilienz.

Skalierbare Anwendungen sind eng mit hochverfügbaren Anwendungen verbunden. In den Kapiteln 7 und 8 haben wir uns lange mit Verfügbarkeitsgruppen und Verfügbarkeitszonen und dem Konfigurieren von Load Balancers beschäftigt. Beide Kapitel drehten sich um die Notwendigkeit, mehrere VMs auszuführen. Wenn Ihre Anwendungen automatisch skalieren können, steigt auch die Verfügbarkeit der Anwendungen, da deren VMs auf Verfügbarkeitsgruppen oder Verfügbarkeitszonen verteilt sind. So weit, so gut. Die Stärke von Azure besteht darin, dass Sie sich keinerlei Gedanken darüber machen müssen, wie Sie weitere Anwendungsinstanzen hinzufügen, diese auf Rechenzentrumshardware oder sogar Rechenzentren verteilen und dann die Netzwerkressourcen aktualisieren, um den Datenverkehr auf die neuen Anwendungsinstanzen zu verteilen.

9.1.1 Vertikales Skalieren von VMs

Das ersten Verfahren beim Skalieren von Ressourcen bildet häufig der althergebrachte Weg, den Sie womöglich in der Vergangenheit genutzt haben. Was würden Sie normalerweise tun, wenn Ihre Anwendung langsamer wird, weil sie von mehr Kund*innen genutzt wird? Die Anzahl der CvPU oder den Arbeitsspeicher vergrößern. Richtig? Sie skalieren die Ressource als Reaktion auf die Nachfrage *vertikal*.

Einer der häufigsten Anwendungsfälle der vertikalen Skalierung sind Datenbankserver. Wenn es um Computerressourcen geht, sind Datenbanken bekanntermaßen hungrig – noch hungriger als Ihre Pizzeria-Kund*innen! Datenbankserver verbrauchen häufig sämtliche Ressourcen, die für eine VM verfügbar sind, selbst wenn sie diese nicht sofort nutzen. Das kann es erschweren, die tatsächlichen Anforderungen an das System zu überwachen und zu wissen, wann vertikal skaliert und mehr Ressourcen bereitgestellt werden müssen. Abbildung 9.2 zeigt die typische vertikale Skalierung für einen Datenbankserver, der mehr Ressourcen benötigt.

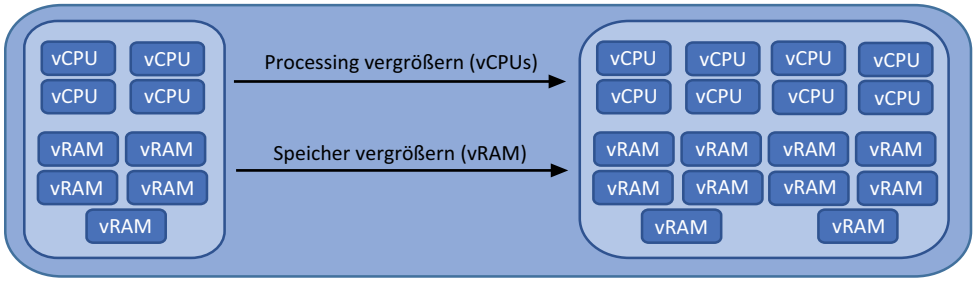


Abbildung 9.2 Wenn eine Datenbank wächst, benötigt sie mehr Ressourcen, um die Daten im Arbeitsspeicher zu speichern und zu verarbeiten. Um in diesem Szenario vertikal zu skalieren, erhöhen Sie die Anzahl der CPUs und die Größe des Arbeitsspeichers.

Möglicherweise müssen Sie über den CPU- oder Arbeitsspeicherbedarf hinaus skalieren. Wie sieht es aus, wenn Sie eine Website betreiben, die viele Bilder oder Videos bereitstellt? Möglicherweise gibt es nicht viele Verarbeitungsanforderungen, aber die Bandbreitenanforderungen sind hoch. Um die verfügbare Bandbreite zu erhöhen, können Sie die Anzahl der NICs Ihrer VM erhöhen. Und wenn Sie mehr Bilder und Videos speichern müssen, dann fügen Sie mehr Speicher hinzu. Sie können Ressourcen hinzufügen oder entfernen, z. B. virtuelle NICs und Speicher, während die VM ausgeführt wird.

Ändern der Grösse virtueller Maschinen

In Azure können Sie die Größe der VM erhöhen (vertikal skalieren), wenn Sie mehr Rechenressourcen für Ihre Anwendung benötigen. In Kapitel 2 hatten Sie eine einfache VM erstellt. Ihre Größe war wahrscheinlich so etwas wie Standard_D2s_v3. Dieser Name sagt Ihnen nicht viel über die Rechenressourcen aus, die einer VM zugewiesen werden. Sie können damit nicht feststellen, ob Sie möglicherweise die CPU oder den Speicher vergrößern müssen. Wenn Sie vertikal skalieren möchten, müssen Sie wissen, welche Optionen Sie haben.

Jetzt testen

Folgen Sie den Anweisungen, um die verfügbaren VM-Größen und Rechenressourcen anzuzeigen:

1. Öffnen Sie das Azure Portal in einem Webbrowser und dann Cloud Shell.
2. Geben Sie den folgenden Azure CLI-Befehl ein, um die verfügbaren VM-Größen und die davon bereitgestellten Rechenressourcen aufzulisten:

```
az vm list-sizes --location eastus --output table
```

Die Ausgabe von `az vm list-sizes` variiert von Region zu Region und ändert sich im Lauf der Zeit, wenn Azure seine VM-Produkte anpasst. Hier sehen Sie eine gekürzte Beispielausgabe, die zeigt, welchen Wert für `MemoryInMb` und `NumberOfCores` die einzelnen VM-Größen bieten:

MaxDataDiskCount	MemoryInMb	Name	NumberOfCores
4	8192	Standard_D2s_v3	2
8	16384	Standard_D4s_v3	4
16	32768	Standard_D8s_v3	8
32	65536	Standard_D16s_v3	16
8	4096	Standard_F2s_v2	2
16	8192	Standard_F4s_v2	4
32	16384	Standard_F8s_v2	8
2	2048	Standard_B1ms	1
2	1024	Standard_B1s	1
4	8192	Standard_B2ms	2
4	4096	Standard_B2s	2

Standard_D2s_v3-VM bietet Ihnen also zwei CPU-Kerne und 8 GB Arbeitsspeicher – mehr als genug für eine einfache VM, auf der ein Webserver läuft. Angenommen, Ihre Online-Pizzeria beginnt, einige Bestellungen zu erhalten, und Sie möchten vertikal skalieren. Sie können mit `az vm resize` eine andere Größe auswählen. Geben Sie dabei die VM-Größe an, die die von Ihrer Anwendung benötigten Werte für die Anzahl CPU-Kerne und die Größe des Arbeitsspeichers bietet.

Die zusätzlichen CPU-Kerne und der Arbeitsspeicher erscheinen nicht wie magisch auf Ihrer VM. Dieses Verhalten kann ein wenig von dem abweichen, was Sie von einer lokalen Hyper-V- oder VMware-Installation kennen. Sie können in einer lokalen Umgebung in gewissem Umfang wesentliche Rechenressourcen hinzufügen oder entfernen, während die VM ausgeführt wird. Wenn Sie die Größe ändern, ist in Azure normalerweise ein VM-Neustart erforderlich, um die neuen Rechenressourcen zu registrieren und die entsprechenden Abrechnungsregeln auszulösen. Wenn Sie vertikal skalieren möchten, müssen Sie daher gewisse Ausfallzeiten einplanen, während die VM neu gestartet wird.

Vertikale Herunterskalierung

Was ist, wenn eine Ihrer VMs mehr Ressourcen hat, als sie benötigt? Dieses Szenario taucht häufiger auf als das einer VM, die weniger Ressourcen hat, als sie benötigt. Der oder die Besitzer*in der Anwendung entscheidet sich vielleicht für eine größere VM, um sicherzustellen, dass seine Anwendung reibungslos läuft. Diese verschwendeten Ressourcen kosten aber Geld und diese Kosten werden oft erst bemerkt, wenn am Monatsende die Rechnung eintrifft.

Ressourcen können in beide Richtungen skaliert werden. Wir haben uns darauf konzentriert, wie Ressourcen *hochskaliert* werden. Dieselben Konzepte gelten aber auch beim *Herunterskalieren* von Ressourcen. Es ist wichtig, die verwendeten VM-Größen zu identifizieren und zu erkennen, wie viel von diesen Ressourcen die Anwendungen nutzen. Sie können dann mit `az vm resize` eine VM-Größe mit weniger CPU-Kernen und Arbeitsspeicher auswählen. Auch für diese resize-Operationen ist zurzeit ein Neustart der VM nötig.

9.1.2 Vertikales Skalieren von Web Apps

Genau wie VMs, können auch Web Apps auf Grundlage der Ressourcenanforderungen hoch- oder herunterskaliert werden. Als Sie in Kapitel 3 eine Web App erstellt haben, stellte die standardmäßige Größe „S1 Standard" einen CPU-Kern und 1,75 GB RAM

bereit. Jede Web App-Stufe und -größe bietet eine festgelegte Anzahl von Ressourcen, wie CPU-Kerne, Arbeitsspeicher und Stagingslots. Selbst wenn sich die Standardgröße oder die Ressourcenzuweisung ändert oder Sie eine andere Web App-Größe auswählen, ändert sich das Konzept nicht.

Wenn Sie Ihre Web App erstellen und feststellen, dass sie mehr Ressourcen benötigt, als der Serviceplan zur Verfügung stellt, können Sie zu einer anderen Stufe wechseln, wie in Abbildung 9.3 dargestellt. Dasselbe gilt, wenn Sie mehr Ressourcen besitzen als Sie benötigen. Sie können Ihre Web App auf diese Weise bei Bedarf manuell hoch- oder herunterskalieren.

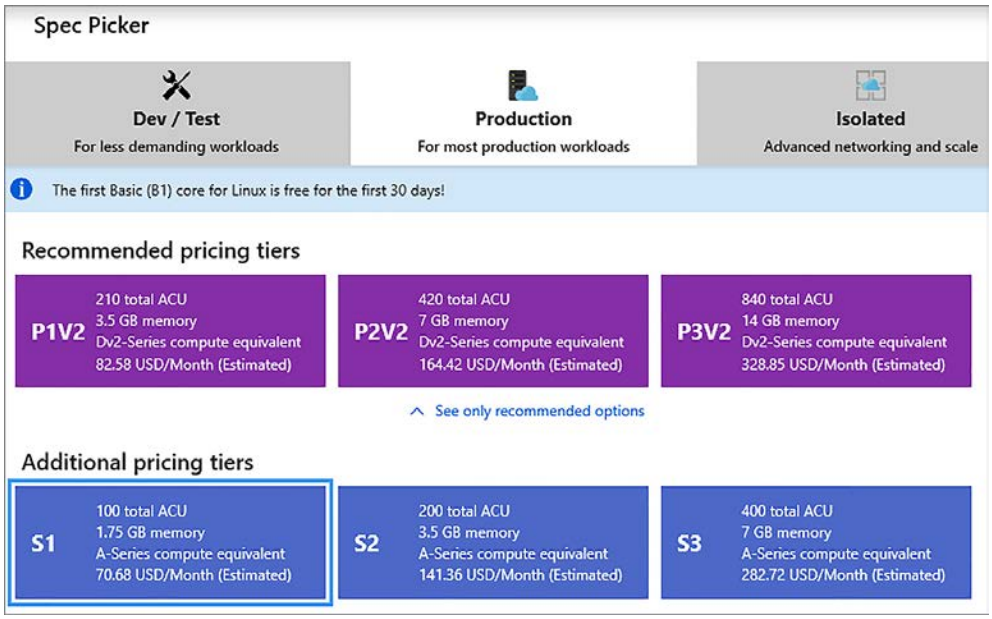


Abbildung 9.3 Um eine Web App manuell vertikal zu skalieren, ändern Sie die Preisstufe (Größe) des zugrunde liegenden App Service Plans. Der App Service Plan definiert den Umfang der Ressourcen, die Ihrer Web-App zugewiesen werden. Wenn Ihre Anwendung einen anderen Umfang an Speicher, Anzahl der CPUs oder Bereitstellungsslots benötigt, können Sie zu einer anderen Stufe wechseln, um die zugewiesenen Ressourcen an die Anforderungen der Anwendung anzupassen.

9.1.3 Horizontales Skalieren von Ressourcen

Ein weiteres Verfahren, um mit der Nachfrage Schritt zu halten, ist die horizontale Skalierung. Um vertikal zu skalieren, erhöhen Sie die Werte für die CPUs und den Speicher, die einer einzigen Ressource zugewiesen werden, zum Beispiel einer VM. Um horizontal zu skalieren, erhöhen Sie stattdessen die Anzahl der VMs, wie in Abbildung 9.4 dargestellt.

Damit sie horizontal skalieren kann, muss sich Ihre Anwendung dieser Fähigkeit bewusst und in der Lage sein, Daten ohne Konflikte zu verarbeiten. Eine Webanwendung ist ein großartiger Kandidat für die horizontale Skalierung, weil die Anwendung Daten in der Regel selbst verarbeiten kann.

Wenn Sie komplexere Anwendungen entwickeln, teilen Sie diese gegebenenfalls in kleinere Einzelkomponenten auf. Wenn Sie an die Azure-Speicherwarteschlange aus Kapitel 4 zurückdenken, haben Sie vielleicht eine Anwendungskomponente, die die

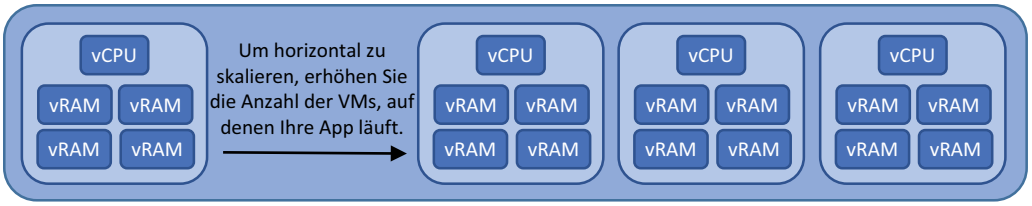


Abbildung 9.4 Um der steigenden Nachfrage nach Ihrer Anwendung gerecht zu werden, können Sie die Anzahl der VMs erhöhen. Dadurch wird die Last auf mehrere VMs verteilt, statt auf immer größere Einzelinstanz-VMs.

Front-End-Web-Bestellungen erhält, und eine andere Anwendungskomponente, die diese Bestellungen verarbeitet und an die Pizzeria überträgt. Nachrichtenwarteschlangen sind ein Ansatz für das Entwerfen und Schreiben von Anwendungen, die in einer Umgebung arbeiten können, die horizontal skaliert. Mit diesem Ansatz können Sie auch jede Anwendungskomponente separat skalieren und verschiedene VM-Größen oder Web-App-Pläne nutzen, um die Effizienz zu maximieren und Ihre monatlichen Kosten zu senken.

Früher haben Sie vertikal skaliert, weil es einfacher war, einer Anwendung mehr Rechenressourcen bereitzustellen und zu hoffen, dass sie damit dann zufrieden ist. Eine Gruppe von Ressourcen einzurichten und eine Anwendung horizontal zu skalieren, hat sich in der physischen Welt oft als komplex erwiesen. Mit Cloud Computing und Virtualisierung werden die Herausforderungen der horizontalen Skalierung so minimiert, dass oft schneller horizontal skaliert werden kann als vertikal – und dass ohne Ausfallzeiten.

Erinnern Sie sich an den Befehl `az vm resize` weiter oben in diesem Kapitel? Was passiert, wenn die Größenänderung der VM abgeschlossen ist? Die VM wird neu gestartet. Falls es sich um die einzige Instanz Ihrer Anwendung handelt, kann erst wieder jemand darauf zugreifen, wenn sie wieder online ist. Wenn Sie horizontal skalieren und VM-Instanzen hinzufügen, gibt es keine Ausfallzeit – sobald die neuen VMs betriebsbereit sind, beginnen sie damit, Anwendungsanfragen zu bearbeiten. Die Integritätstests des Load Balancers aus Kapitel 8 erkennen automatisch, wenn eine neue VM im Backend-Pool bereit ist, Kundenanfragen zu bearbeiten, woraufhin Datenverkehr an sie verteilt wird.

Azure ist für Flexibilität und Wahlmöglichkeit bei der Skalierung ausgelegt. Wenn Sie eine neue Anwendungsumgebung entwerfen, sollten Sie einen horizontalen Skalierungsansatz implementieren. VMs haben in Azure einen tollen Verwandten, der Ihnen hier weiterhelfen kann: VM-Skalierungsgruppen.

9.2 VM-Skalierungsgruppen

VMs bilden aus gutem Grund eine der häufigsten Workloads in Azure. Die Lernkurve für das Erstellen und Ausführen einer VM ist flach, weil sich fast alles, was Sie bereits wissen, direkt in Azure übertragen lässt. Webserver sind eine der häufigsten Workloads einer VM. Das ist wiederum praktisch, da Sie keine neuen Kompetenzen erwerben müssen, um Ihr Wissen für das Ausführen von Apache, IIS oder NGINX in einem Azure-VM zu übertragen.

Wie sieht es mit einem VM-Cluster aus, in dem ein Webserver läuft? Wie würden Sie das in Ihrer regulären On-Premises-Umgebung handhaben? Zunächst gibt es viele mögliche Clusterlösungen. Wie sieht es mit Updates für Ihre physischen Server oder VMs aus? Wie würden Sie damit umgehen? Was wäre, wenn Sie die Anzahl der Instanzen im Cluster automatisch erhöhen oder verringern wollten? Brauchen Sie dafür ein weiteres Tool? In Abbildung 9.5 sehen Sie eine Übersicht einer VM-Skalierungsgruppe.

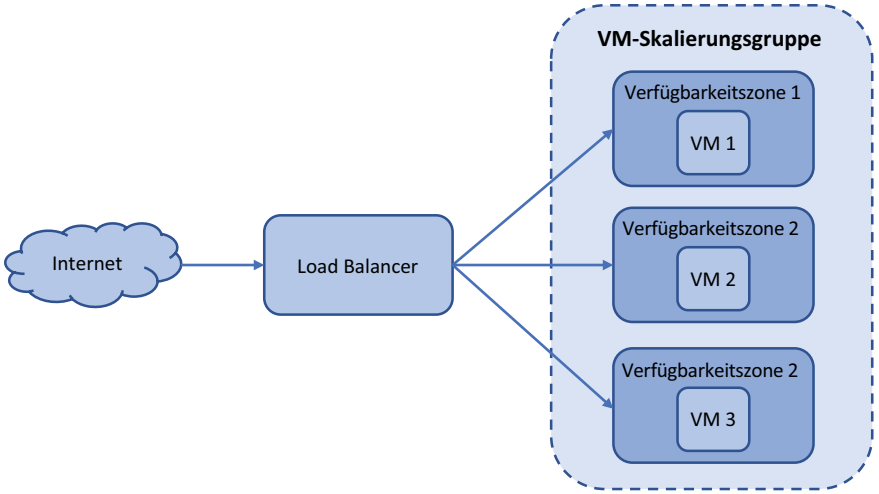


Abbildung 9.5 Eine VM-Skalierungsgruppe sorgt für eine logische Gruppierung von VMs. Die VMs sind identisch und können zentral verwaltet, aktualisiert und skaliert werden. Sie können Metriken definieren, die die Anzahl der VMs in der Skalierungsgruppe auf Basis der Workload Ihrer Anwendung automatisch vergrößert oder verkleinert.

Eine Skalierungsgruppe vereinfacht das Ausführen und Verwalten mehrerer VMs für die Bereitstellung einer hochverfügbaren Anwendung mit Lastausgleich. Sie teilen Azure mit, welche VM-Größe, welches Basisimage für die VM und wie viele Instanzen Sie verwenden wollen. Sie können dann CPU- oder Speichermetriken definieren, um die Anzahl der Instanzen in Reaktion auf die Anwendungslast oder abhängig von einem Zeitplan zu den Stoßzeiten automatisch zu erhöhen oder zu verringern. Skalierungsgruppen kombinieren das IaaS-Modell für VMs mit der Leistung von PaaS-Funktionen wie Skalierung, Redundanz, Automatisierung und zentraler Verwaltung von Ressourcen.

Eine VM-Skalierungsgruppe für eine einzige VM?

Wenn Sie Anwendungen auf VMs aufbauen, planen Sie von Anfang an mit einer Skalierungsgruppe, auch wenn Sie nur eine VM benötigen. Warum? Eine Skalierungsgruppe kann jederzeit erweitert werden und stellt automatisch die Verbindungen zu einem Load Balancer oder Application Gateway her. Wenn die Nachfrage nach der Anwendung in zwei Monaten plötzlich ansteigt, können Sie die Skalierungsgruppe anweisen, eine oder zwei zusätzliche VM-Instanzen zu erstellen.

Um eine normale, eigenständige VM zu erweitern, müssen Sie diese VM zu einem Load Balancer hinzufügen. Und wenn Sie die VM nicht von Anfang an in einer Verfügbarkeitsgruppe oder Verfügbarkeitszone geplant haben, müssen Sie sich überlegen, wie Sie diese VMs hochverfügbar machen können. Indem Sie gleich eine Skalierungsgruppe erstellen – selbst für eine einzige VM –, machen Sie Ihre Anwendung mit minimalem Mehraufwand zukunftssicher.

9.2.1 Erstellen einer VM-Skalierungsgruppe

Auch wenn eine Skalierungsgruppe das Entwickeln und Ausführen hochverfügbarer Anwendungen vereinfacht, müssen Sie ein paar neue Komponenten erstellen und konfigurieren. Abgesehen davon, können Sie den Prozess der Bereitstellung einer Skalierungsgruppe mit der Azure CLI auf zwei Befehle reduzieren.

Jetzt testen

Führen Sie die folgenden Schritte aus, um eine Skalierungsgruppe mit der Azure CLI zu erstellen:

1. Öffnen Sie das Azure Portal, und klicken Sie oben im Dashboard auf das Cloud Shell-Symbol.
2. Erstellen Sie eine Ressourcengruppe mit `az group create`, legen Sie einen Namen fest, zum Beispiel `azuremolchapter9`, und geben Sie einen Standort ein:

   ```
   az group create --name azuremolchapter9 --location westeurope
   ```

   Skalierungsgruppen können Verfügbarkeitszonen nutzen, stellen Sie daher sicher, dass Sie eine Region auswählen, in der diese verfügbar sind.
3. Um eine Skalierungsgruppe zu erstellen, geben Sie die Anzahl der gewünschten VM-Instanzen an und wie die VM-Instanzen Aktualisierungen ihrer Konfiguration behandeln sollen. Wenn Sie an den VMs eine Änderung vornehmen, indem Sie zum Beispiel eine Anwendung installieren oder Updates des Gastbetriebssystems durchführen, können sich die VMs automatisch aktualisieren, sobald sie die Änderung erkennen. Sie können für die Upgraderichtlinie aber auch manual festlegen und die Updates zu einem geeigneten Zeitpunkt anwenden. Die restlichen Parameter sollten Sie bereits vom Erstellen einer einzelnen VM kennen:

   ```
   az vmss create \
     --resource-group azuremolchapter9 \
     --name scalesetmol \
     --image UbuntuLTS \
     --admin-username azuremol \
     --generate-ssh-keys \
     --instance-count 2 \
     --vm-sku Standard_B1ms \
     --upgrade-policy-mode automatic \
     --lb-sku standard \
     --zones 1 2 3
   ```

So einfach ist das. Sie haben mehrere VMs in einer Verfügbarkeitszone erstellt, die skalieren kann. Und die Skalierungsgruppe, die Sie gerade mit der Azure CLI erstellt haben, bietet Ihnen ein wirklich tolles Feature. Erinnern Sie sich an das vorherige Kapitel über Load Balancer und die ganzen CLI-Befehle, die Sie verwenden mussten, und die Suche nach Vorlagen, um das Erstellen des Load Balancers zu vereinfachen? Der Befehl `az vmss create` hat für Sie bereits einen Load Balancer erstellt und konfiguriert!

Denken Sie an Ihre Kontingentgrenzen

Ich habe dieses Kontingentproblem in Kapitel 7 bereits erwähnt, es lohnt sich aber, es zu wiederholen, falls Probleme auftreten. In Azure verhindern Standardkontingente für Ihr Abonnement, dass Sie versehentlich Ressourcen bereitstellen und dann vergessen, was Geld kostet! Die Liste der Kontingente finden Sie unter http://mng.bz/ddcx.

Wenn Sie mehrere VMs erstellen, können Kontingentprobleme auftreten. Probleme können auch dann auftreten, wenn Sie die Ressourcen aus früheren Kapiteln und Übungen nicht löschen. Wenn Sie eine Fehlermeldung wie

```
Operation results in exceeding quota limits of Core.
Maximum allowed: 4, Current in use: 4, Additional requested: 2.
```

sehen, ist dies ein gutes Zeichen dafür, dass Sie eine Erhöhung Ihrer Kontingente beantragen müssen. Sie können Ihr aktuelles Kontingent für eine bestimmte Region wie folgt anzeigen:

```
az vm list-usage --location westeurope
```

Führen Sie die unter http://mng.bz/Xq2f beschriebenen Schritte aus, um eine Erhöhung des Kontingents für eine Region anzufordern.

Die Azure CLI hilft Ihnen, mit wenigen Eingabeaufforderungen eine Skalierungsgruppe zu erstellen. Ein Load Balancer wurde erstellt und konfiguriert, eine öffentliche IP-Adresse zugewiesen und die VM-Instanzen der Skalierungsgruppe wurden zum Back-End-IP-Pool hinzugefügt.

Jetzt testen

Sehen Sie sich die mit Ihrer Skalierungsgruppe erstellten Ressourcen an, die im Folgenden beschrieben werden.

Führen Sie den folgenden Befehl aus, um anzuzeigen, welche Ressourcen mit Ihrer Skalierungsgruppe erstellt wurden:

```
az resource list \
  --resource-group azuremolchapter9 \
  --output table
```

Ihre Ausgabe ähnelt dem folgenden Beispiel. Sehen Sie sich in Spalte „Type" an, ob ein virtuelles Netzwerk, eine öffentliche IP-Adresse und ein Load Balancer erstellt wurden:

Name	ResourceGroup	Type
mol	azuremolchapter9	Microsoft.Compute/virtualMachineScaleSets
molLB	azuremolchapter9	Microsoft.Network/loadBalancers
molLBIP	azuremolchapter9	Microsoft.Network/publicIPAddresses
molVNET	azuremolchapter9	Microsoft.Network/virtualNetworks

Was bedeutet dieser ganze Zauber? Wenn Sie mit der Azure CLI eine Skalierungsgruppe erstellen, werden für Sie ein zonenredundanter Load Balancer und eine öffentliche IP-Adresse erstellt. Die VMs werden erstellt und zu einem Backend-IP-Pool auf dem Load Balancer hinzugefügt. NAT-Regeln werden erstellt, die Ihnen die Verbindung mit den VM-Instanzen ermöglichen. Es fehlen einzig Load Balancer-Regeln, da diese je nach ausgeführter Anwendung variieren. Wenn Sie VMs zur Skalierungsgruppe hinzufügen

oder daraus entfernen, wird automatisch die Konfiguration des Load Balancers aktualisiert, um die Verteilung des Datenverkehrs an die neuen Instanzen zuzulassen. Diese Zauberei beschränkt sich nicht nur auf die Azure CLI – auch wenn Sie die Azure PowerShell oder das Azure Portal verwenden, werden diese unterstützenden Netzwerkressourcen erstellt und miteinander verbunden, um zusammenzuarbeiten.

Jetzt testen

Ihre Skalierungsgruppe wurde mit zwei Instanzen erstellt. Sie können die Anzahl der VM-Instanzen in Ihrer Skalierungsgruppe manuell skalieren. Der Load Balancer aktualisiert dann automatisch die Konfiguration des Backend-IP-Pools. Legen Sie mit `--new-capacity` für die Skalierungsgruppe wie folgt vier Instanzen fest:

```
az vmss scale \
    --resource-group azuremolchapter9 \
    --name scalesetmol \
    --new-capacity 4
```

9.2.2 Erstellen von Regeln für die automatische Skalierung

Als Sie Ihre Skalierungsgruppe erstellt haben, haben Sie eine feste Anzahl von Instanzen bereitgestellt. Eine der herausragendsten Eigenschaften von Skalierungsgruppen ist die Möglichkeit, die Anzahl der von der Skalierungsgruppe ausgeführten VM-Instanzen automatisch hoch- oder herunterzuskalieren.

Wie in Abbildung 9.6 dargestellt, kann die Anzahl der Instanzen in einer Skalierungsgruppe bei zunehmender Anwendungslast automatisch erhöht werden. Denken Sie an eine typische Geschäftsanwendung in Ihrer Umgebung. Zu Beginn des Arbeitstags beginnen die Benutzer*innen damit, auf die Anwendung zuzugreifen, was dazu führt, dass die Ressourcenbelastung dieser VM-Instanzen steigt. Um eine optimale Anwendungsleistung sicherzustellen, fügt die Skalierungsgruppe automatisch weitere VM-Instanzen hinzu. Der Load Balancer beginnt automatisch, den Datenverkehr an die neuen Instanzen zu verteilen. Später am Tag, wenn die Benutzer*innen nach Hause gehen, sinkt die Nachfrage nach Anwendungen. Die VM-Instanzen verbrauchen weniger Ressourcen, sodass die Skalierungsgruppe automatisch einige VM-Instanzen entfernt, um unnötige Ressourcen zu reduzieren und die Kosten zu senken.

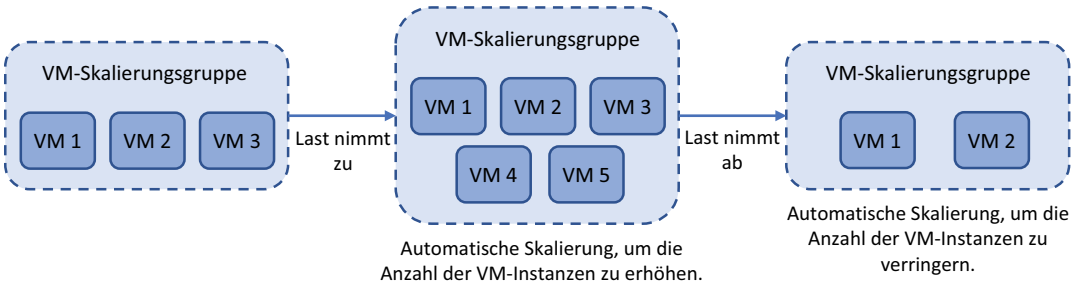


Abbildung 9.6 Skalierungsgruppen können automatisch hoch- und herunterskalieren. Sie definieren Regeln, um bestimmte Kennzahlen zu überwachen, die Regeln zum Erhöhen oder Reduzieren der Anzahl der ausgeführten VM-Instanzen auslösen. Die Anzahl der VM-Instanzen ändert sich, sobald sich die Nachfrage nach Ihrer Anwendung ändert. Durch diesen Ansatz wird die Leistung und Verfügbarkeit Ihrer Anwendung maximiert. Gleichzeitig werden unnötige Kosten minimiert, sobald die Anwendungslast sinkt.

Sie können Ihre Regeln für Skalierungsgruppen auf verschiedene Metriken stützen. Sie können sich die Host-Metriken für den grundlegenden Ressourcenverbrauch ansehen, die Metriken für Gast-VMs für die Analyse bestimmter Leistungsindikatoren für Anwendungen konfigurieren oder mit Azure Application Insights den Anwendungscode im Detail überwachen.

Mit Zeitplänen kann auch für ein Zeitfenster eine bestimmte Anzahl von VM-Instanzen in einer Skalierungsgruppe definiert werden. Im Beispiel einer allgemeinen Geschäftsanwendung, bei der die Nachfrage während der Geschäftszeiten höher ist als am Abend, möchten Sie vielleicht eine höhere feste Anzahl von Instanzen für die Geschäftszeiten und eine niedrigere feste Anzahl von Instanzen für die Abendstunden festlegen.

Autoskalierungsregeln, die auf Metriken basieren, überwachen die Leistung in einem definierten Zeitintervall, z. B. 5 Minuten, und benötigen dann weitere Minuten, um die neuen VM-Instanzen zu starten und für die Anwendung zu konfigurieren. Wenn Sie feste Zeitpläne verwenden, um die Anzahl der VM-Instanzen in Ihrer Skalierungsgruppe automatisch zu skalieren, sind diese zusätzlichen Ressourcen bereits in Gebrauch und der Load Balancer verteilt den Datenverkehr während des Tags an diese Instanzen.

Für die Verwendung von Zeitplänen ist ein Ausgangswert für die typische Anwendungsanforderung erforderlich. Eine höhere oder niedrigere Nachfrage an bestimmten Stellen des Geschäftszyklus wird nicht berücksichtigt. Letztlich besteht die Gefahr, dass Sie teilweise mehr Ressourcen haben, als Sie brauchen, und so mehr als nötig bezahlen. Und es können Situationen auftreten, in denen die Anwendungslast das übersteigt, was die VM-Instanzen in der Skalierungsgruppe bewältigen können.

Jetzt testen

Führen Sie die folgenden Schritte aus, um die Autoskalierungsregeln für eine Skalierungsgruppe zu erstellen:

1. Suchen Sie in der Navigationsleiste links im Azure Portal nach „Resource Group" (Ressourcengruppe), und wählen Sie diese Option aus.
2. Wählen Sie die Ressourcengruppe aus, die Sie für die Bereitstellung Ihrer Vorlage erstellt haben, z. B. `azuremolchapter9`.
3. Wählen Sie in der Liste der Ressourcen Ihre Skalierungsgruppe aus, z. B. „skalesetmol".
4. Wählen Sie unter „Settings" (Einstellungen) links im Fenster „Scale Set" (Skalierungsgruppe) und dann „Scaling" (Skalierung) aus. Sie können manuell skalieren oder Ihre eigenen Regeln für die automatische Skalierung erstellen.
5. Wählen Sie diese Option, um benutzungsdefinierte Regeln für die automatische Skalierung zu erstellen.
6. Geben Sie einen Namen ein, z. B. `autoscale`, und definieren Sie dann eine minimale, maximale und standardmäßige Anzahl an Instanzen. Legen Sie für diese Übung für das Minimum 2, das Maximum 10 und für den Standardwert 2 fest.
7. Klicken Sie auf „Add a Rule" (Regel hinzufügen), und sehen Sie sich die verfügbaren Regeleinstellungen an, wie in Abbildung 9.7 dargestellt.

   Die Standardparameter richten sich nach dem durchschnittlichen CPU-Verbrauch. Die Regel löst aus, wenn die Last in einem Intervall von 10 Minuten über 70 % liegt. Die Skalierungsgruppe wird um eine VM-Instanz erhöht, und die Regeln warten dann 5 Minuten, bevor sie mit der Überwachung beginnen und gegebenenfalls die nächste Regel auslösen.

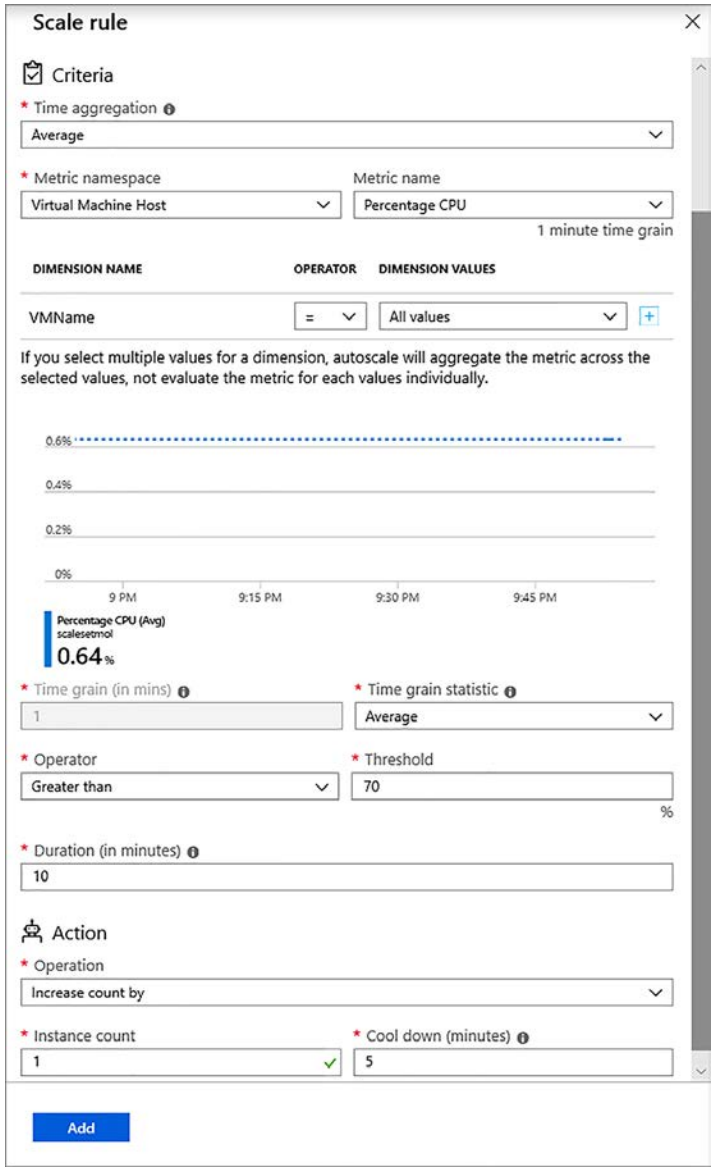


Abbildung 9.7 Wenn Sie eine Autoskalierungsregel hinzufügen, definieren Sie das genaue Verhalten, das für das Auslösen der Regel erforderlich ist.

Während dieser Cool-Down-Phase haben die neuen VM-Instanzen Zeit für ihre Bereitstellung und können beginnen, den Datenverkehr vom Load Balancer zu empfangen. Dadurch sollte die Gesamtlast in der Skalierungsgruppe sinken. Ohne diese Cool-Down-Phase könnten die Regeln das Hinzufügen einer weiteren VM-Instanz auslösen, bevor die Last an die zuvor erstellte VM-Instanz verteilt wird.

8 Klicken Sie auf „Add" (Hinzufügen), um die Regel zu erstellen.

9 Fügen Sie eine weitere Regel hinzu. Konfigurieren Sie die Regel dieses Mal so, dass die Anzahl um eins reduziert wird, wenn die durchschnittliche CPU-Last für eine Dauer von 5 Minuten unter 30 % liegt.

10 Überprüfen Sie Ihre Regeln, wie in Abbildung 9.8 dargestellt, und klicken Sie dann auf „Save" (Speichern).

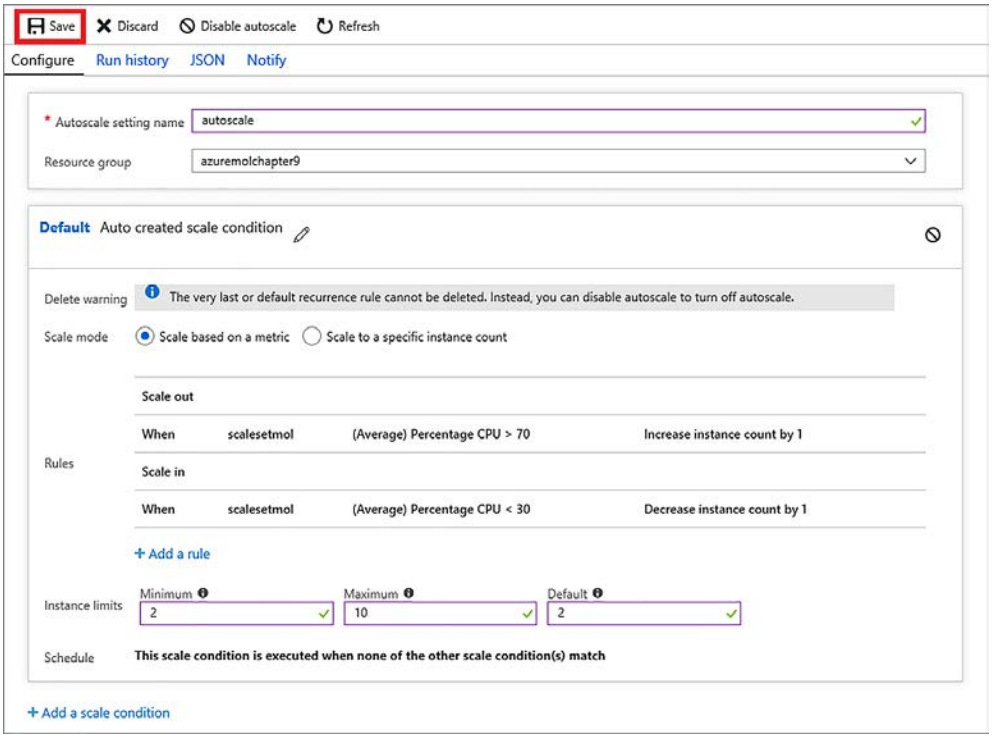


Abbildung 9.8 Sie sollten jetzt eine Regel besitzen, die die Anzahl der Instanzen um eins erhöht, wenn die durchschnittliche CPU-Last über 70 % liegt. Und eine weitere Regel, die die Anzahl der Instanzen um eins verringert, wenn die durchschnittliche CPU-Last unter 30 % liegt.

Sie können Autoskalierungsregeln auch mit der Azure CLI, Azure PowerShell oder über Templates konfigurieren. Das Portal bietet eine schöne visuelle Möglichkeit, Regeln zu überprüfen und die verfügbaren Optionen für die einzelnen Parameter anzuzeigen. Wenn Sie komplexere Regeln erstellen, bieten Vorlagen eine Möglichkeit, Skalierungsgruppen mit demselben Regelwerk auf reproduzierbare Weise zu erstellen.

9.3 Skalieren einer Web App

Falls Sie sich sehr für Web Apps in Kapitel 3 interessiert haben oder für Azure-Tabellen und -Warteschlangen in Kapitel 4, können diese letzten drei Kapitel, die den Schwerpunkt auf IaaS-VMs gelegt haben, anstrengend gewesen sein. Sollte die Cloud nicht einfacher sein? Für PaaS-Komponenten wie Web Apps gilt das absolut!

Ich möchte nicht, dass Sie das Gefühl haben, dass wir die nächsten Seiten, auf denen es um die Hochverfügbarkeit und die Autoskalierung von Web Apps geht, eventuell schnell abhaken wollen. Die Wahrheit lautet, dass dies viel einfacher ist! Wie meistens entscheidet die Wahl zwischen IaaS und PaaS über den Ausgleich zwischen Flexibilität und unkomplizierter Verwaltung. Bei PaaS-Services, wie Web Apps, wird ein großer Teil der zugrunde liegenden Redundanz abstrahiert. Daher ist kein ganzes Kapitel über Hochverfügbarkeit und ein weiteres Kapitel über Load Balancer erforderlich.

Die Entscheidung für IaaS für die Entwicklung und den Betrieb eigener VMs oder Skalierungsgruppen mit Load Balancern und Verfügbarkeitszonen kann aufgrund geschäftlicher Anforderungen oder einer Beschränkung getroffen werden. Entwickler*innen, Betriebsingenieur*innen oder Tools und Workflows sind möglicherweise nicht vollständig auf Web Apps vorbereitet. Dennoch rate ich Ihnen dringend, Web Apps für neue Anwendungsbereitstellungen zu berücksichtigen. Die Nutzung von PaaS-Komponenten, wie Web Apps, gibt Ihnen mehr Zeit, sich auf die Apps und Ihre Kund*innen zu konzentrieren, als auf die Infrastruktur und die Verwaltung.

Jetzt testen

Führen Sie die folgenden Schritte aus, um mit der Azure CLI eine Web App zu erstellen:

1. In Kapitel 3 haben Sie eine Web App im Azure Portal erstellt. Wie bei den meisten Ressourcen ist es oft schneller und einfacher, die Azure CLI zu verwenden. Öffnen Sie die Cloud Shell im Azure Portal.
2. Erstellen Sie einen App Service Plan mit der Größe „S1 Standard". Mit dieser Größe können Sie automatisch auf bis zu zehn Instanzen Ihrer Web App skalieren:

```
az appservice plan create \
  --name appservicemol \
  --resource-group azuremolchapter9 \
  --sku s1
```

3. Erstellen Sie eine Web App, die ein lokales Git-Repository für die Bereitstellung verwendet, wie Sie es bereits in Kapitel 3 gemacht haben:

```
az webapp create \
  --name webappmol \
  --resource-group azuremolchapter9 \
  --plan appservicemol \
  --deployment-local-git
```

Alle in Abschnitt 9.2.2 über Autoskalierungsregeln und Zeitpläne für Skalierungsgruppen besprochenen Konzepte und Szenarien gelten auch für Web Apps. Als kurzen Rückblick finden Sie hier einige gängige Szenarien für die Autoskalierung von Web Apps:

- Automatisches Vergrößern oder Verringern der Anzahl der Web App-Instanzen anhand von Leistungsmetriken, um die wechselnde Nachfrage während des gesamten Arbeitstages zu unterstützen.
- Planen einer Web App, die die Anzahl der Instanzen zu Beginn des Arbeitstags automatisch erhöht und dann am Ende des Arbeitstags verringert.

Im Fall der Pizzeria erhält die Web App später am Tag und am Abend möglicherweise mehr Datenverkehr. Daher passt ein Satz Autoskalierungsregeln nicht für alle Situationen. Auch hier brauchen Sie einen Ausgangswert für Ihre Anwendungsleistung, um zu verstehen, wie sie bei normaler Verwendung läuft und bei welcher Leistungsmetrik die App hoch- oder herunterskalieren muss. Auch wenn Sie Autoskalierungszeitpläne verwenden, sollten Sie weiterhin überwachen und verfolgen, wann es zu Nachfragespitzen kommt, um Regeln zu erstellen, die dieses Nutzungsmuster unterstützen.

Jetzt testen

Führen Sie die folgenden Schritte aus, um die Autoskalierungsregeln für eine Web App zu erstellen:

1. Suchen Sie in der Navigationsleiste links im Azure Portal nach „Resource Group" (Ressourcengruppe), und wählen Sie diese Option aus.
2. Wählen Sie die Ressourcengruppe aus, die Sie für Ihre Web App erstellt haben, z. B. „azuremolchapter9".
3. Wählen Sie in der Liste der Ressourcen Ihre Web App aus, z. B. „webappmol".
4. Wählen Sie unter „Settings" (Einstellungen) links im Web App-Fenster „Scale Out (App Service Plan)" (Horizontal hochskalieren (App Service Plan)) aus.
5. Auch hier können Sie benutzerdefinierte Regeln für die automatische Skalierung konfigurieren und nicht nur die Web App manuell skalieren.
6. Geben Sie einen Namen ein, z. B. `autoscalewebapp`, und definieren Sie dann die minimale, maximale und standardmäßige Anzahl der Instanzen. Legen Sie für diese Übung für das Minimum 2, das Maximum 5 und für den Standardwert 2 fest.
7. Klicken Sie auf „Add a Rule" (Regel hinzufügen), und sehen Sie sich die verfügbaren Regeleinstellungen an. Dieses Fenster sieht genauso aus wie das für die Autoskalierungsregeln für Skalierungsgruppen. Die Standardparameter richten sich nach dem durchschnittlichen CPU-Verbrauch und lösen aus, wenn die Last in einem 10-Minuten-Intervall über 70 % liegt. Die Web App wird um eine Instanz erhöht. Die Regeln warten dann 5 Minuten, bevor sie mit der Überwachung beginnen und gegebenenfalls die nächste Regel auslösen.
8. Fügen Sie eine weitere Regel hinzu. Konfigurieren Sie die Regel dieses Mal so, dass die Anzahl um eins reduziert wird, wenn die durchschnittliche CPU-Last für eine Dauer von 5 Minuten unter 30 % liegt.
9. Überprüfen und speichern Sie Ihre Regeln.

Wenn Ihre Autoskalierungsregeln das Hoch- oder Herunterskalieren der Web App auslösen, aktualisiert die Azure-Plattform die Verteilung des Datenverkehrs an die verfügbaren Web App-Instanzen. Sie haben keinen Load Balancer wie bei Skalierungsgruppen, aber der Datenverkehr wird trotzdem automatisch auf die Web App-Instanzen verteilt, wenn Sie Ihre Umgebung nach oben oder unten skalieren. Das Konzept ist ähnlich, nur von Ihnen abstrahiert, denn Sie sollen den PaaS-Ansatz genießen und sich nicht so viele Gedanken machen!

Sowohl Skalierungsgruppen als auch Web Apps bieten eine Möglichkeit, Regeln zu erstellen, die automatisch die Anzahl der Instanzen skalieren, die Ihre Anwendungen ausführen. Mit mehreren Instanzen, die Ihre Anwendung ausführen, erhöhen Sie auch die Verfügbarkeit Ihrer App. Skalierungsgruppen bilden einen guten Mittelweg für Entwickler*innen und Manager*innen, die Anwendungen auf VMs aufbauen wollen oder müssen, während sie PaaS-ähnliche Funktionen nutzen, um den Kundendatenverkehr automatisch zu skalieren und neu zu konfigurieren.

In Kapitel 11 befassen wir uns mit dem Azure Traffic Manager, der diese Hochverfügbarkeitsbereitstellungen wirklich vervollständigt. Im Moment sind Sie noch nicht ganz produktionsreif, wenn es darum geht, mehrere redundante Skalierungsgruppen oder Web App-Instanzen anzubieten, auf die der Datenverkehr automatisch verteilt wird. Wir kommen gleich noch darauf zurück.

9.4 Übung: Installation von Anwendungen in Ihrer Skalierungsgruppe oder der Web App

Wir haben in diesem Kapitel vieles behandelt. Jetzt können Sie sich für eine kurze abschließende Übung für Skalierungsgruppen oder Web Apps entscheiden. Oder für beide, falls Sie Ihre Mittagspause verlängern möchten!

9.4.1 VM-Skalierungsgruppen

Sie haben in Ihren Skalierungsgruppen mehrere VM-Instanzen, die im Moment aber nicht viel machen. Eine Übersicht der verschiedenen Verfahren für die Installation von Anwendungen in VM-Instanzen in einer Skalierungsgruppe finden Sie unter http://mng.bz/9Ocx. In der Praxis würden Sie eine der automatisierten Bereitstellungsmethoden anwenden. Installieren Sie jetzt aber manuell einen Webserver in den VM-Instanzen, wie Sie es in Kapitel 8 getan haben:

1. Erinnern Sie sich an die NAT-Regeln des Load Balancers? Standardmäßig besitzt jede VM-Instanz in einer Skalierungsgruppe eine NAT-Regel, die Ihnen eine direkte SSH-Verbindung zur Instanz ermöglicht. Die Ports liegen nicht am Standard-TCP-Port 22. Die Liste der VM-Instanzen in einer Skalierungsgruppe und deren Portnummern können Sie wie folgt anzeigen:

```
az vmss list-instance-connection-info \
  --resource-group azuremolchapter9 \
  --name scalesetmol
```

2. Benutzen Sie den Parameter `-p` wie folgt, um sich über SSH mit einem bestimmten Port zu verbinden (geben Sie Ihre eigene öffentliche IP-Adresse und Ihre Portnummern an):

```
ssh azuremol@40.114.3.147 -p 50003
```

3. Installieren Sie auf jeder VM-Instanz mit `apt install` einen einfachen NGINX-Webserver. Denken Sie daran, wie Sie in Kapitel 8 vorgegangen sind.
4. Um die Skalierungsgruppe in Aktion zu sehen, öffnen Sie die öffentliche IP-Adresse des Load Balancers der Skalierungsgruppe in einem Webbrowser.

5 Wenn Sie Probleme haben, vergewissern Sie sich, dass der Load Balancer eine Load Balancer-Regel für TCP-Port 80 korrekt erstellt hat und einen zugehörigen Integritätstest entweder für TCP-Port 80 oder Ihren eigenen benutzungsdefinierten HTTP-Integritätstest hat, der auf der VM nach /health.html sucht.

9.4.2 Web Apps

Der Prozess zum Bereitstellen Ihrer Anwendung für eine Web App, die mehrere Instanzen ausführt, stimmt mit dem für eine einzelne Web Apps in Kapitel 3 überein. Sie übertragen die Anwendung per Push in das lokale Git-Repository für die Web App und die Azure-Plattform stellt diese einzelne Codebasis dank PaaS in mehreren Web App-Instanzen bereit:

1 Initialisieren Sie ein Git-Repository in „azure-mol-samples-2nd-ed/09", und fügen Sie dann die Beispieldateien hinzu, und committen Sie sie, wie Sie es in Kapitel 3 getan haben:

```
cd azure-mol-samples-2nd-ed/09
git init && git add . && git commit -m "Pizza"
```

2 Ihre Web App verfügt über ein lokales Git-Repository. Fügen Sie auf die gleiche Weise eine Remote-Verbindung für Ihre Web App hinzu, wie Sie es in Kapitel 3 getan haben:

```
git remote add webappmolscale <your-git-clone-url>
```

3 Übertragen Sie dieses Beispiel per Push in Ihre Web App. Mit dieser einzelnen Codeübertragung wird Ihre App an die unterschiedlichen Web App-Instanzen verteilt:

```
git push webappmolscale master
```

10 Globale Datenbanken mit Cosmos DB

Daten. Es führt kein Weg an ihnen vorbei. Nahezu jede eine Anwendung, die Sie entwickeln und ausführen, erstellt und verarbeitet Daten oder ruft sie ab. Diese Daten wurden bisher meist in einer strukturierten Datenbank wie MySQL, Microsoft SQL oder PostgreSQL gespeichert. Derartige große, strukturierte Datenbanken sind etabliert und wohlbekannt. Es gibt umfangreiche Dokumentationen und Tutorials darüber und ein Zugriff darauf ist mit den meisten großen Programmiersprachen möglich.

Je größer die Möglichkeiten, umso größer ist die Verantwortung. Aufwand und Verwaltung der Infrastruktur erfolgen in der Regel über diese traditionell strukturierten Datenbanken. Es spricht bei weitem nichts dagegen, sie zu verwenden. Doch bei Anwendungen auf globaler Ebene, ist es nicht einfach, Cluster von Datenbankservern zu erstellen, die Ihre Daten replizieren und Kund*innen intelligent in Ihre nächste Instanz leiten.

An dieser Stelle kommt Azure Cosmos DB ins Spiel. Sie brauchen sich nicht darum zu kümmern, wie Sie Ihre Daten replizieren, die Konsistenz gewährleisten und Kundenanfragen verarbeiten können. Stattdessen füttern Sie eines der vielen verfügbaren Modelle mit Ihren Daten und wählen aus, wo diese Daten zur Verfügung stehen sollen. Dieses Kapitel behandelt die Grundlagen unstrukturierter Datenbankmodelle bei Cosmos DB. Sie erfahren, wie Sie Ihre Datenbank für die globale Distribution erstellen und konfigurieren können und wie Sie Webanwendungen entwickeln, die Ihre äußerst redundante und skalierbare Cosmos DB-Instanz nutzen.

10.1 Was ist Cosmos DB?

In Kapitel 4 ging es um unstrukturierte Datenbanken mit Azure-Speichertabellen. Auch wenn das verwendete Beispiel recht einfach war, sind die dort verwendeten Begriffe grundlegend für Cosmos DB. Betrachten wir zunächst noch einmal den Unterschied zwischen *strukturierten* und *unstrukturierten* Datenbanken.

10.1.1 Strukturierte Datenbanken (SQL)

Strukturierte Datenbanken sind die traditionellere Methode, Daten zu speichern. Die *Struktur* oder das *Schema* einer Datenbank legen fest, wie Daten dargestellt werden. Daten werden in Tabellen gespeichert. Jede Zeile steht für ein Element und eine Gruppe von Werten, die diesem zugeordnet sind. Kommen wir auf das Modell der Pizzeria zurück: In jeder Tabellenzeile, in der Pizzasorten gespeichert sind, können Angaben über die Bezeichnung, die Größe und den Preis der Pizza stehen. Eine einfache SQL-Datenbank ist in Abbildung 10.1 dargestellt.

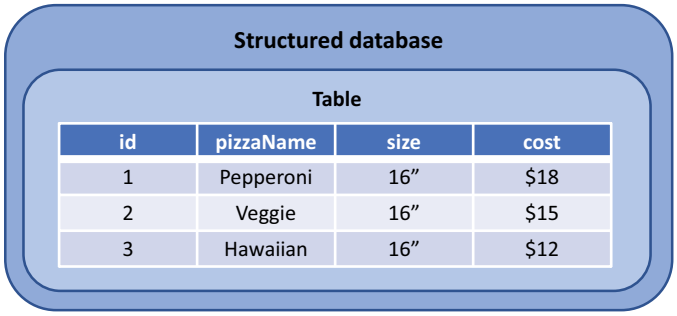
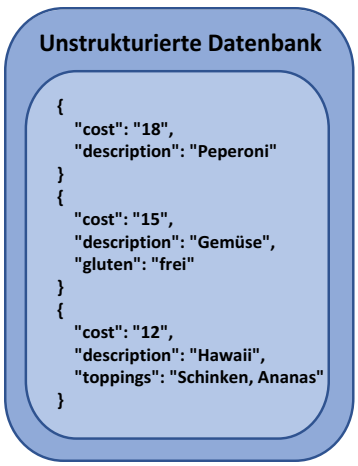


Abbildung 10.1 In einer strukturierten Datenbank werden Daten innerhalb einer Tabelle in Zeilen und Spalten gespeichert. Jede Zeile enthält eine festgelegte Reihe an Spalten. Sie bilden das Datenbankschema.

Damit Datenabfragen in strukturierten Datenbanken erfolgreich vonstatten gehen können, muss jeder Server in der Regel die gesamte Datenbank enthalten. Die Daten werden in Abfragen zusammengefügt, die anhand von Kriterien, die der oder die Entwickler*in als Teil der strukturierten Abfrage erstellt, aus verschiedenen Tabellen abgerufen werden. Daher stammt der Begriff *Structured Query Language* (SQL). Datenbanken werden immer größer und komplexer. Daher müssen die Server, auf denen eine Datenbank läuft, ausreichend groß sein, um die Daten im Speicher zu verarbeiten. Bei sehr großen Datensätzen ist das eine schwierige und kostspielige Angelegenheit. Ist zudem eine Struktur vonnöten, wird es außerdem schwierig, Eigenschaften hinzuzufügen und die Struktur später zu verändern.

10.1.2 Unstrukturierte Datenbanken (NoSQL)



Die unstrukturierten Daten in NoSQL-Datenbanken werden nicht in Tabellen mit Zeilen und Spalten gespeichert, sondern in dynamischen Arrays. So können Sie nach Bedarf neue Eigenschaften für ein Element hinzufügen. Diese Methode hat den großen Vorteil, dass Sie schnell eine neue Pizzasorte oder einen speziellen Belag hinzufügen können, ohne die zugrundeliegende Datenbankstruktur zu ändern. Bei einer strukturierten Datenbank müssen Sie eine neue Spalte zu einer Tabelle hinzufügen und dann die Anwendung aktualisieren, um die zusätzliche Spalte zu bearbeiten. Bei NoSQL-Datenbanken fügen Sie eine weitere Eigenschaft zu einem bestimmten Eintrag aus Ihrem Code hinzu (siehe Abbildung 10.2.).

Abbildung 10.2 In unstrukturierten Datenbanken werden Daten ohne feste Zuordnungen von Spalten in einer Tabellenzeile gespeichert. Sie können etwa Beläge zu einer einzelnen Pizza hinzufügen, ohne das gesamte Datenbankschema und andere Datensätze aktualisieren zu müssen.

NoSQL-Datenbanken bieten auch verschiedene Datenbankmodelle. Diese Modelle geben Aufschluss darüber, wie die Daten gespeichert und in der Datenbank abgerufen werden. Die Größe und das Format der Daten, mit denen Sie arbeiten, bestimmen das zu verwendende Modell. Entscheidend ist zudem, wie Sie die Daten in Ihrer Anwendung darstellen müssen. Diese Modelle enthalten Dokumente, Grafiken und Tabellen. Kümmern Sie sich erst einmal nicht zu sehr um die Modelle. Je nachdem, welche Relationen und Abfragen Sie für Ihre Daten benötigen, funktionieren unterschiedliche Modelle besser für verschiedene Sätze unstrukturierter Daten. Der Knackpunkt ist, dass unstrukturierte NoSQL-Datenbanken die Daten prinzipiell anders speichern und abrufen. Dies kann vorteilhaft für Sie sein, wenn Sie Cloud-Anwendungen in Azure erstellen und ausführen.

10.1.3 Skalieren von Datenbanken

Erinnern Sie sich daran, dass bei strukturierten Datenbanken die gesamte Datenbank in der Regel auf jedem Server existieren muss? Sehr große Datenbanken erfordern folglich immer größere Server. Auch wenn Ihre Datenbanken niemals auf Größenordnungen von Hunderten Gigabyte oder gar Terabyte anwachsen, dürfte es Sie interessieren, dass die Erweiterung und Skalierung von NoSQL-Datenbanken anders erfolgt als bei SQL-Datenbanken. Der Unterschied besteht darin, dass die Skalierung bei NoSQL-Datenbanken eher horizontal als vertikal erfolgt.

Die vertikale Skalierung einer VM durch Aufrüsten von Arbeitsspeicher und Prozessorkapazitäten hat ihre Grenzen. Wenn Sie den maximalen Speicherdurchsatz und die Netzwerkbandbreite voll ausnutzen, kommt es in anderen Bereichen des Compute-Stacks zu Leistungsproblemen. Ganz zu schweigen von den Kosten derart großer VMs, die auf Sie (oder Ihren Chef) zukommen werden. Abbildung 10.3 verdeutlicht noch einmal die in Kapitel 9 behandelte vertikale Skalierung. Stellen Sie sich nun den Cluster derart großer Datenbank-VMs vor, schließlich soll Ihre Anwendung ja auch über Redundanz und Elastizität verfügen.

Bei einer horizontalen Skalierung hingegen, können Sie Datenbank-VMs mit weniger Ressourcen und zu einem niedrigeren Preis betreiben. Zu diesem Zweck verteilen NoSQL-Datenbanken die Daten über Datenbankknoten und leiten Anfragen von Ihrer Anwendung auf den entsprechenden Knoten. Die anderen Knoten im Cluster müssen nicht wissen, wo alle Daten gespeichert sind.

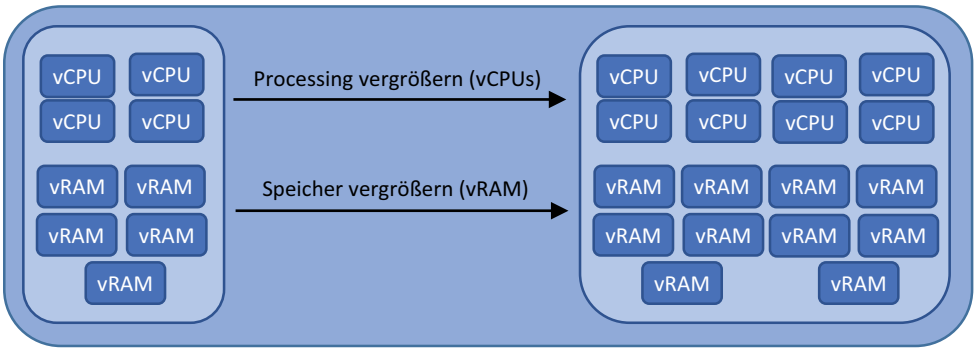


Abbildung 10.3 Die Skalierung traditionell strukturierter Datenbanken erfolgt vertikal. Größere Datenbanken benötigen mehr Speicherplatz, mehr Arbeitsspeicher und eine stärkere Prozessorleistung auf dem Server.

Sie müssen lediglich auf ihre eigenen Anfragen zu reagieren. Knoten lassen sich in Reaktion auf eine Kundennachfrage schnell zu einem Cluster hinzufügen.

Daher muss die gesamte Datenbank in einer NoSQL-Datenbank nicht in den Speicher eines Hosts passen. Sie brauchen nur einen Teil der Datenbank, einen *Shard*, zu speichern und verarbeiten. Bei Anwendungen mit großen Mengen *strukturierter* Daten kann eine NoSQL-Datenbank die Leistung beeinträchtigen, da die Informationen bei verschiedenen Hosts abgerufen werden müssen. NoSQL-Datenbanen können zu einer besseren Performance sowie Management- und Effizienzvorteilen führen, wenn große Mengen *unstrukturierter* Daten zu verarbeiten sind. Ein Beispiel für die horizontale Skalierung unstrukturierte Datenbanken über mehrere Hosts finden Sie in Abbildung 10.4.

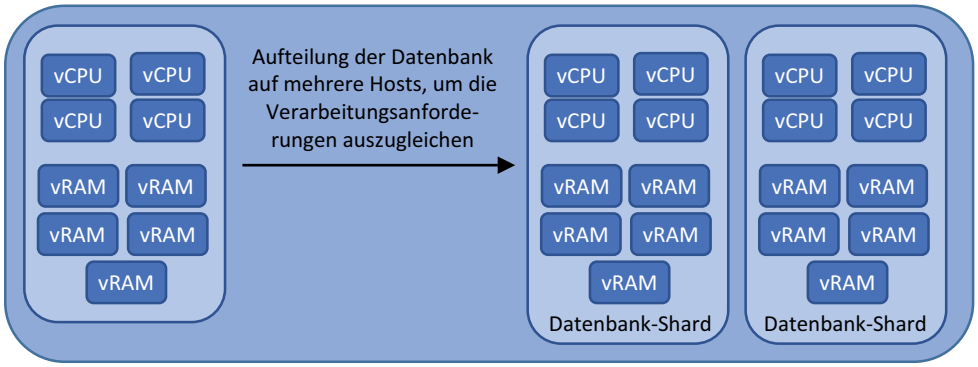


Abbildung 10.4 Die Skalierung unstrukturierter NoSQL-Datenbanken erfolgt horizontal. Wenn eine Datenbank wächst, wird sie in Datensegmente aufgesplittet, die auf die Datenbankserver verteilt werden.

10.1.4 Mit Cosmos DB alles zusammenbringen

Was ist also Cosmos DB? Cosmos DB ist eine automatisch skalierende, global verteilte Datenbankplattform, die es Ihnen ermöglicht, verschiedene Formen von NoSQL-Datenbanken zu nutzen. Cosmos DB nimmt Ihnen, ähnlich wie Web Apps und vergleichbare Dienste, viel Managementarbeit ab. Wenn Sie eine Web App erstellen, müssen Sie weder den Lastausgleich noch die Cluster-Bildung konfigurieren. Wählen Sie einfach Ihre Regionen aus, konfigurieren Sie die automatische Skalierung und laden Sie Ihren Anwendungscode hoch. Die Azure-Plattform repliziert und verteilt den Datenverkehr der Web App so, dass eine hohe Verfügbarkeit gewährleistet ist. Wenn Sie Cosmos DB verwenden, brauchen Sie sich keine Gedanken darüber zu machen, wie hoch Ihre Datenbankanforderungen sind,wie viel Arbeitsspeicher Sie zuweisen sollten oder mit welcher Methode Ihre Daten für die Redundanz zu replizieren sind. Sie wählen lediglich den erforderlichen Datendurchsatz und die Region für die Datenspeicherung aus und können dann Ihre Daten hinzufügen.

Dieses Kapitel verwendet ein SQL-Modell für Cosmos DB. Die Daten werden allerdings in dem NoSQL-Format JSON gespeichert. Lassen Sie sich nicht irritieren, wenn dies neue Begriffe für sie sind. Auch andere Modelle wie Mongo, Cassandra, Gremlin und Table können verwendet werden. Die Funktionalität ist für alle gleich: Wählen Sie Ihr Modell und Ihre Regionen aus, und fügen Sie Ihre Daten hinzu. Das ist der Vorteil von Cosmos DB.

10.2 Erstellen eines Cosmos DB-Kontos und einer Datenbank

Sehen wir uns Cosmos DB und unstrukturierte Datenbanken in der Praxis an. Dazu haben wir verschiedene Möglichkeiten: Sie können über das Azure Portal ein Konto erstellen und anschließend ein Datenbankmodell auswählen und erstellen. Anschließend geben Sie Daten in die Datenbank ein, die Ihre App abfragen kann. Alternativ können Sie Azure CLI, Azure PowerShell oder sprachspezifische Software Development Kits (SDKs) verwenden, um alles im Code zu erstellen. Wir verwenden hier das Azure Portal, damit Sie die Daten auch visuell erstellen und abfragen können.

10.2.1 Erstellen einer Datenbank unter Cosmos DB und Befüllung mit Daten

Im Kapitel 4 haben Sie mithilfe einer Azure-Speichertabelle Ihre erste NoSQL-Datenbank erstellt. Nun verwenden wir Cosmos DB, um eine ähnliche Datenbank zu erstellen. Doch diese Datenbank enthält alle Geo-Redundanz- und Replikationsmöglichkeiten, die Ihren Kund*innen ohne Ausfallzeiten die Bestellung einer Pizza über Ihren Online-Shop ermöglichen. Zunächst erstellen Sie ein Cosmos DB-Konto und eine Dokumentendatenbank. Anschließend geben Sie einige Datensätze für drei Pizzasorten hinzu (siehe Abbildung 10.5.).

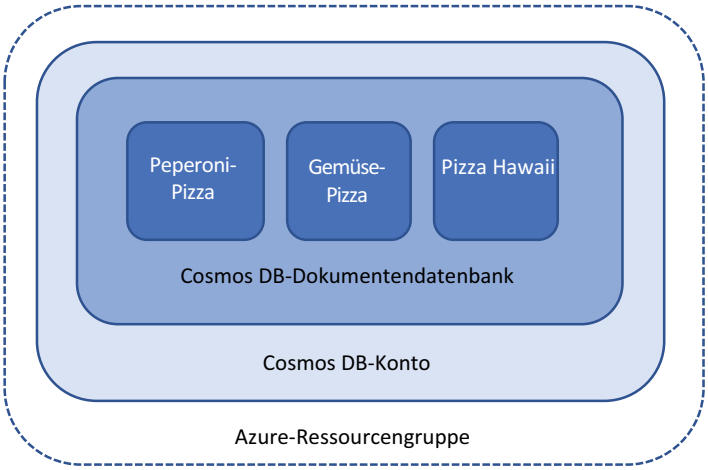


Abbildung 10.5 In diesem Abschnitt erstellen Sie eine Ressourcengruppe und ein Cosmos DB-Konto. In diesem Konto wird dann eine Dokumentendatenbank erstellt. Sie fügen drei Datensätze hinzu, die eine einfache Speisekarte für Ihre Pizzeria darstellen.

Jetzt testen

Erstellen Sie über das Azure Portal ein Konto, um Cosmos DB in Aktion zu sehen:

1. Öffnen Sie das Azure Portal und wählen Sie oben links im Dashboard „Create a Resource“ (Eine Ressource erstellen) aus.
2. Suchen und wählen Sie Azure Cosmos DB, und klicken Sie anschließend auf „Create“ (Erstellen).
3. Erstellen Sie eine Ressourcengruppe, z. B. `azuremolchapter10`, und geben Sie einen eindeutigen Namen für Ihr Cosmos DB-Konto ein, z. B. `azuremol`.

4. Der für Ihre Datenbank verwendbare Modelltyp wird als API bezeichnet. Wählen Sie bei unserem Beispiel im Dropdown-Menü „Core (SQL)“ aus.
5. Wählen Sie als Region USA Ost aus. Cosmos DB ist für alle Azure-Regionen verfügbar. In diesem Beispiel erwartet die Webanwendung, die Sie in der Übung am Ende des Kapitels bereitstellen jedoch, dass Sie die Region USA Ost verwenden.
6. Lassen Sie die Option für die Georedundanz deaktiviert, ebenso wie alle anderen zusätzlichen Funktionen, z. B. Multiwrite-Regionen. In Abschnitt 10.2.2 gehen wir näher darauf ein, wie Sie Ihre Datenbank global replizieren können.

Sicherer Datenverkehr mit Service-Endpunkten

Sie haben die Möglichkeit, Ihre Cosmos DB über einen sogenannten *Dienstendpunkt* mit einem virtuellen Azure-Netzwerk zu verbinden. Wir werden diese Option jetzt nicht diskutieren, aber es ist eine coole Funktion, die dazu beiträgt, Ihre Instanz zu sichern, indem sie den Zugriff auf die Datenbank nur von einem bestimmten virtuellen Netzwerk aus erlaubt.

Wenn Sie Middleware-Anwendungen erstellen, die Cosmos DB verwenden, oder nur interne Anwendungen, können Sie einen Endpunkt für virtuelle Netzwerkdienste verwenden, um den Zugriff von einem bestimmten virtuellen Netzwerk aus einzuschränken, nicht über das Internet und mit einem öffentlichen Endpunkt. Eine wachsende Zahl von Azure-Diensten unterstützt diese Art von Endpunkten, und dies ist ein weiteres Beispiel dafür, dass Sie die Möglichkeit haben, Ihre Umgebung so zu sichern, dass sie Ihren Geschäftsanforderungen entspricht.

7. Wenn Sie bereit sind, überprüfen und erstellen Sie Ihr Cosmos DB-Konto. Es dauert einige Minuten, das VM-Konto zu erstellen.

Ihre Datenbank ist noch leer. Wie können Sie nun einige Daten für die Speisekarte Ihrer Pizzeria speichern? Cosmos DB gruppiert die Daten in einer Datenbank in etwas, das als *Container* bezeichnet wird. Nein, das ist nicht dieselbe Art von Container, die die treibende Kraft hinter Docker, Kubernetes und Cloud-Native-Anwendungen ist, von denen Sie vielleicht schon gehört haben. Diese Namensverwirrung ist kein Pluspunkt, aber bleiben Sie vorerst bei mir.

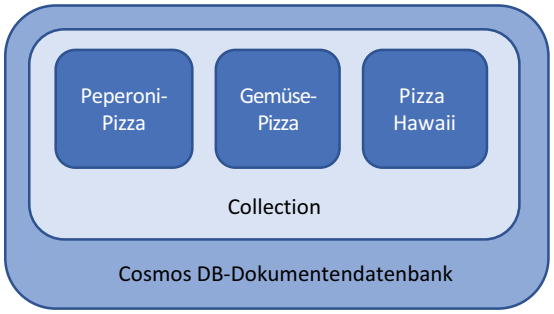


Abbildung 10.6 Cosmos DB-Datenbanken, die das Dokumentenmodell verwenden, speichern Daten in Sammlungen. Dank dieser Sammlungen können Sie Daten für eine schnellere Indexierung und Abfrage gruppieren.

Bei Cosmos DB-Datenbanken, die das Dokumentenmodell verwenden, werden Daten logisch in Containern, so genannten *Sammlungen* gruppiert. Andere API-Modelle haben einen etwas anderen Namen für die Container-Entität, z. B. *graph* für die Gremlin-API. Für unsere SQL-API werden Sammlungen verwandter Daten gespeichert, die schnell indiziert und abgefragt werden können (siehe Abbildung 10.6.). Sammlungen ähneln in gewisser Weise einer traditionellen, in Tabellen organisierten SQL-Datenbank, doch sie sind hinsichtlich einer auf Performance oder Redundanz bezogenen Datenverteilung sehr viel flexibler.

Cosmos DB ist für sehr große Datenmengen und einen hohen Datendurchsatz ausgelegt. Sie können daher entscheiden, wie Sie den Datenfluss und die damit verbundenen Kosten bemessen und steuern wollen. Der Durchsatz wird in Anforderungseinheiten pro Sekunde (RU/s) berechnet. Eine Anforderungseinheit entspricht 1 KB Dokumentendaten. Im Wesentlichen legen Sie die gewünschte Bandbreite für Ihre Datenbank fest. Je mehr Bandbreite (RU/s) Sie beanspruchen, umso mehr zahlen Sie. Cosmos DB zeigt Ihnen an, wie viele Daten Sie nutzen und wie hoch der Durchsatz Ihrer Anwendung ist. In der Regel brauchen Sie sich nicht allzu große Mühe zu geben, die richtigen Größenordnungen zu finden. Für Ihr Beispiel der Pizzeria wollen wir keine allzu hohen Ansprüche stellen!

Jetzt testen

Befolgen Sie die Schritte in Abbildung 10.9, um eine Sammlung zu erstellen und einige Einträge in die Datenbank einzugeben.

1. Suchen Sie in der Navigationsleiste links im Azure Portal nach „Resource Group" (Ressourcengruppe), und wählen Sie diese Option aus.
2. Wählen Sie die Ressourcengruppe aus, in der Sie Ihre Cosmos DB-Datenbank erstellt haben, wie beispielsweise azuremolchapter10.
3. Wählen Sie Ihr Cosmos DB-Konto aus der Liste der Ressourcen aus, und gehen Sie dann auf die Übersichtsseite.
4. Fügen Sie einen Container hinzu.
5. Da dies Ihre erste Datenbank ist, wählen Sie Neu erstellen. Geben Sie einen Namen ein, wie zum Beispiel `pizzadb`.
6. Belassen Sie den Durchsatz auf dem Standardwert.
7. Geben Sie für die Container-ID `Pizzen` ein. So erstellen Sie einen logischen Container, in dem Sie die in Ihrer Pizzeria angebotenen Artikel speichern können.
8. Geben Sie einen Partitionsschlüssel `/Beschreibung` ein, um sicherzustellen, dass Pizza-Typen gleichmäßig verteilt werden.

   Der Partitionsschlüssel gibt an, wie die Daten in der Datenbank aufgeteilt werden können. Es ist in einer kleinen Beispieldatenbank wie dieser nicht wirklich erforderlich, aber sie zu verwenden, ist eine bewährte Vorgehensweise, wenn Ihre App skaliert.
9. Fügen Sie keinen eindeutigen Schlüssel hinzu. Mithilfe von Schlüsseln können Sie einen Container enger logisch definieren und zum Beispiel Untergruppen für Bestellungen festlegen. Die größere Sammlung ist für Ihre Speisekarte gedacht. In viel größeren Datenbänken können Sie Schlüssel aber auch für Pizzen, Getränke und Desserts in Kategorien aufteilen.
10. Wählen Sie OK, um die Datenbank und die Sammlung zu erstellen.

Nun verfügen Sie über ein Cosmos DB-Konto, eine Datenbank und eine Sammlung, aber Cosmos DB enthält noch keine Pizzen. Sie haben die Möglichkeit, Daten zu importieren oder einen Code für die Eingabe größerer Datenmengen zu schreiben. Wir geben nun manuell drei Pizzen ein, damit Sie sehen können, wie die in das Azure Portal integrierten grafischen Such- und Abfragetools funktionieren und wie Sie die Daten in Ihrer Cosmos DB-Datenbank bearbeiten können.

Jetzt testen

Befolgen Sie die Schritte in Abbildung 10.7, um einige Einträge in die Datenbank einzugeben.

1 Wählen Sie aus dem Menü links im Übersichtsfenster Ihres Cosmos DB-Kontos Daten-Explorer aus.

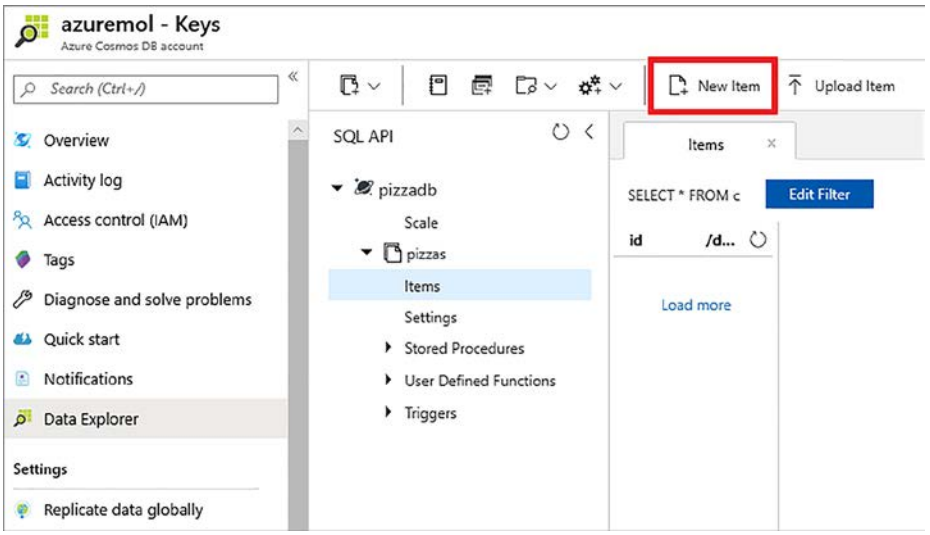


Abbildung 10.7 Mit dem Daten-Explorer im Azure Portal können Sie Ihre Sammlungen durchsuchen, um Abfragen zu tätigen oder neue Dokumente zu erstellen. Mit diesem grafische Tool verwalten Sie Ihre Datenbank schnell von einem Webbrowser aus.

2 Erweitern Sie die Datenbank `pizzadb` und anschließend die Sammlung `pizzas`.

3 Fügen Sie ein neues Element hinzu, um einige Pizzen in die Datenbank zu legen. Die Daten werden im JSON-Format hinzugefügt.

4 Ersetzen Sie in einem Textfeld bestehenden Text mit den folgenden Daten, um ein neues Menüelement für eine normale Pepperonipizza zu erstellen:

```
{
    "description": "Pepperoni",
    "cost": "18"
}
```

5 Gehen Sie auf Speichern, um die Daten zur Datenbank hinzuzufügen.

6 Fügen Sie eine weitere Pizza zu Ihrer Speisekarte hinzu. Geben Sie dieses Mal eine Eigenschaft ein, die darüber informiert, dass es sich um eine Pizza aus glutenfreiem Teig handelt. Sie brauchen dazu nichts Besonderes in der verwendeten Datenbank zu tun. Fügen Sie einfach eine weitere Eigenschaft zu Ihren Daten hinzu. Um ein weiteres neues Element hinzuzufügen, geben Sie die folgenden Daten ein, und wählen Sie „Save“ (Speichern):

```
{
    "description": "Veggie",
    "cost": "15",
    "gluten": "free"
}
```

7 Fügen Sie eine weitere Pizzasorte hinzu. Geben Sie dieses Mal eine Eigenschaft ein, die darüber informiert, womit die Pizza belegt ist. Um ein weiteres neues Element hinzuzufügen, geben Sie die folgenden Daten ein, und wählen Sie „Save“ (Speichern):

```
{
    "description": "Hawaiian",
    "cost": "12",
    "toppings": "ham, pineapple"
}
```

Diese drei Einträge verdeutlichen, was mit einer NoSQL-Datenbank alles möglich ist. Sie haben Eigenschaften zu den Datensätzen hinzugefügt, ohne dass Sie das Datenbankschema aktualisieren mussten. Zwei verschiedene Eigenschaften informieren darüber, dass die vegetarische Pizza aus glutenfreiem Teig besteht und womit die Pizza Hawaii belegt ist. Cosmos DB akzeptiert diese zusätzlichen Eigenschaften. Diese Daten stehen nun in Ihren Anwendungen zur Verfügung.

Einige zusätzliche JSON-Eigenschaften werden für Dinge wie `id`, `_rid` und `_self` hinzugefügt. Dies sind keine Eigenschaften, für die Sie sich kümmern müssen. Cosmos DB verwendet diese Eigenschaften, um die Daten nachzuverfolgen und zu identifizieren. Sie sollten Sie nicht manuell bearbeiten oder löschen.

10.2.2 Versehen einer Cosmos DB-Datenbank mit globaler Redundanz

Sie verfügen nun über eine Cosmos DB-Datenbank, in der eine einfache Pizza-Speisekarte in der Region USA Ost gespeichert ist. Aber Sie möchten auf der ganzen Welt Konzessionen für Ihre Pizzeria vergeben! Sie möchten die Daten zu Ihren Pizzen in Azure-Regionen an verschiedenen Standorten in die Nähe Ihrer neuen Kund*innen replizieren.

Wozu dient dieser Schritt? Wenn alle Ihre Kund*innen Daten aus der Datenbank in einer Region lesen und schreiben müssen, entsteht ein erheblicher potenzieller Datenverkehr durch Kabel unter dem Meer, der um die ganze Welt geleitet wird. Damit Ihre Kund*innen möglichst geringe Wartezeiten erleben, können Sie Ihre Daten in Azure-Regionen auf der ganzen Welt replizieren. Ihre Kund*innen werden mit dem nächstgelegenen Replikat verbunden (siehe Abbildung 10.8).

Die Cosmos DB-Plattform enthält Konsistenzebenen mit entsprechenden Garantien, welche die Datenkonsistenz und Replikation für Sie erledigen. Sie bestimmen eine oder mehrere Regionen als primäre Schreibregion. Die Beispiele in diesem Buch gehen von einer einzigen Schreibregion aus. Mithilfe der Multimaster-Funktion können Sie Ihre Daten jedoch auf den nächstgelegenen Endpunkt schreiben. Dieser wird dann übertragen.

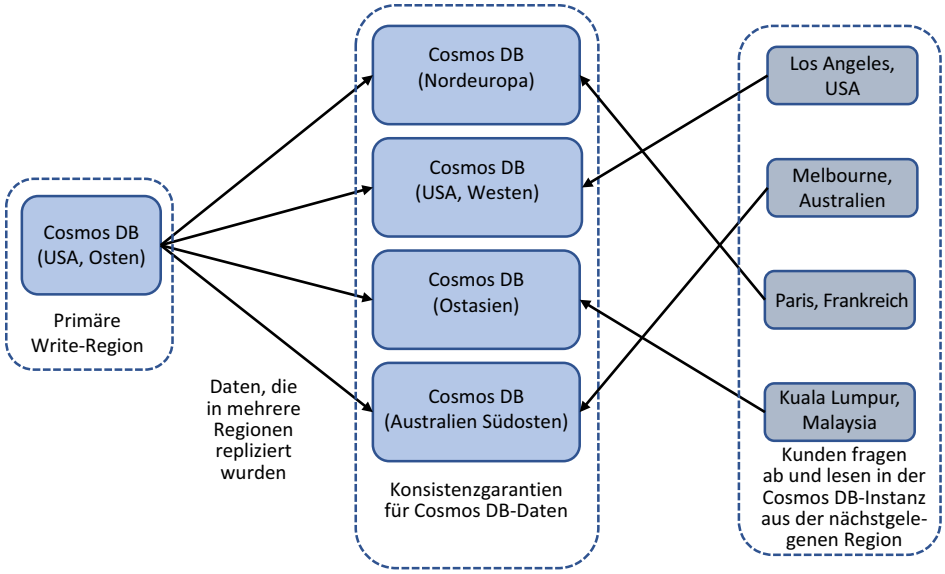


Abbildung 10.8 Die Daten werden von einer primären Cosmos DB-Instanz auf mehrere Azure-Regionen in der ganzen Welt repliziert. Webanwendungen lassen sich so konfigurieren, dass Daten aus der den Kund*innen am nächsten gelegenen Region gelesen und der Datenverkehr mit ihnen dynamisch dorthin geleitet wird. Minimale Latenz und bessere Reaktionszeiten sind die Folge.

asynchron in andere Regionen. Die Daten werden außerdem schnell auf die von Ihnen bestimmten Leseregionen repliziert. Sie können die Failover-Reihenfolge kontrollieren, um Leseregionen festzulegen. Außerdem können Sie die gewünschten Leseregionen mit Ihrer Anwendung automatisch oder manuell bestimmen.

Sie können eine Konsistenzebene festlegen, die definiert, wie schnell Schreibvorgänge über mehrere Regionen repliziert werden. Dies betrifft eher gestalterische als betriebliche Aspekte. Die Konsistenzebenen reichen von *strong*, was abgewartet, bis die replizierten Schreibvorgänge von den Replikaten bestätigt werden. Dies garantiert die Konsistenz der Replikate. *Letztliche* Konsistenz ist weniger streng. Das letztendliche Modell garantiert, dass alle Daten repliziert werden, aber es kann zu einer leichten Verzögerung kommen, wenn Lesevorgänge von Replikaten unterschiedliche Werte zurückgeben, bis sie alle synchronisiert sind.

Es besteht ein ausgewogenes Verhältnis zwischen einer eingeschränkteren geografischen Verteilung, wie zum Beispiel bei der starken Konsistenzebene und einer weiter gefassten geografischen Replikation bei der letztlichen Konsistenzebene, allerdings vor dem Hintergrund, dass es bei der Datenreplikation leichte Verzögerungen geben kann. Je nachdem, wie konsistent und zeitnah die Datenreplikation erfolgen soll, entstehen Kosten für Bandbreite und Bearbeitung. Die Azure-Plattform übernimmt die betreffende Datenreplikation von Ihrer Schreibregion aus. Sie müssen weder Anwendungen für die Datenreplikation entwickeln, noch festlegen, wie die Daten aus den replizierten Endpunkten am besten zu lesen sind.

Das bedeutet auf globaler Ebene: Sie können mehrere VM oder Web Apps, die Sie in den vorherigen Kapiteln entwickelt haben, in verschiedenen Regionen auf der ganzen Welt nutzen. Diese Apps verbinden sich für Abfragen und das Lesen aller Daten mit einer lokalen Cosmos DB-Instanz. Im nächsten Kapitel behandeln wir einige hilfreiche Azure Netzwerkverkehr-Features, mit denen Anwender*innen automatisch auf eine lokale Webanwendungsinstanz geleitet werden können, die auch eine lokale Cosmos DB-Instanz verwenden. Bei regionalen Ausfällen oder Wartungsarbeiten leitet die gesamte Plattform den Kund*innen auf die nächstgelegene Instanz.

Ein solches Setup erfordert in der traditionell strukturierten Welt der Datenbanken, in der VMs zu verwalten, Datenbanken zu installieren und Cluster zu konfigurieren sind, eine ausgefeilte Entwurfsplanung. Seine Implementierung ist recht kompliziert. Mit Cosmos DB erledigen Sie alles mit drei Mausklicks. Ehrlich!

Jetzt testen

Führen Sie folgende Schritte aus, um Ihre Cosmos DB-Daten global zu replizieren:

1. Suchen Sie in der Navigationsleiste links im Azure Portal nach „Resource Group" (Ressourcengruppe), und wählen Sie diese Option aus.
2. Wählen Sie die Ressourcengruppe aus, in der Sie Ihre Cosmos DB-Datenbank erstellt haben, wie beispielsweise `azuremolchapter10`.
3. Wählen Sie aus der Liste aller Ressourcen Ihr Cosmos DB-Konto aus. Diese beiden Mausklicks waren kostenlos, aber ab jetzt tickt die Uhr!
4. Wählen Sie die Menüoption auf der linken Seite aus, um Daten global zu replizieren. Die Karte mit allen verfügbaren Azure-Regionen zeigt, dass Ihre Datenbank derzeit in der Region USA Ost verfügbar ist (Abbildung 10.9).

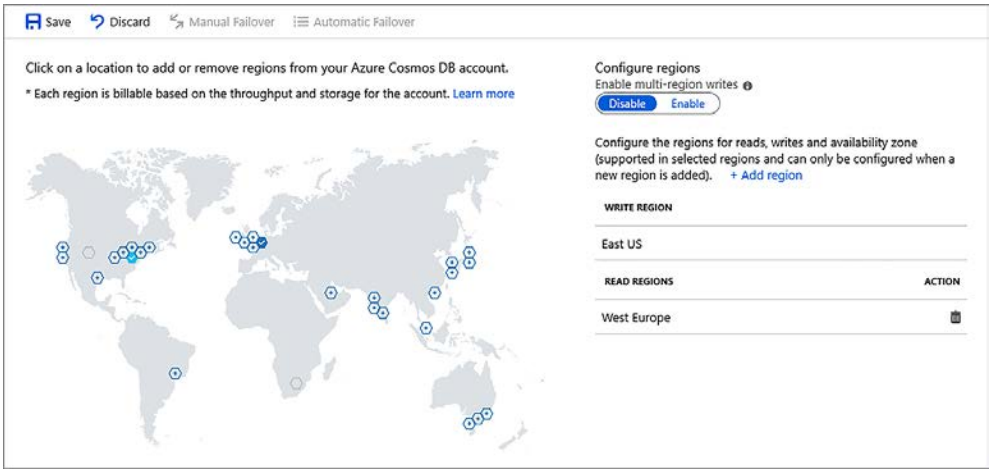


Abbildung 10.9 Wählen Sie eine Azure-Region für die Replikation Ihrer Cosmos DB-Datenbank aus, und gehen Sie auf „Save" (Speichern). Diese Schritte sind erforderlich, um Ihre Daten global zu verteilen.

5 Wählen Sie Westeuropa und dann „Save“ (Speichern). Sie können jede gewünschte Azure-Region auswählen, doch die Übung am Ende des Kapitels verlangt, dass Ihre Daten nach Westeuropa repliziert werden. Es dauert etwas, bis die Daten in die von Ihnen gewählte Region repliziert und für Ihre Anwendung online gestellt werden.

Zählen Sie die Mausklicks! Drei Klicks, oder? Seien wir großzügig und zählen die ersten beiden Mausklicks mit, die Sie benötigt haben, um die Ressourcengruppe und das Cosmos DB-Konto auszuwählen. Mit nicht mehr als fünf Mausklicks und in wenigen Sekunden haben Sie eine Replikat-Instanz Ihrer Datenbank erstellt, die es Ihren Anwendungen ermöglicht, aus der nächstgelegenen Region auf Ihre Daten zuzugreifen. Wäre so etwas mit einem MySQL-Cluster möglich? Bitte schreiben Sie auf Twitter an @fouldsy, wenn es Ihnen gelingt, diesen Vorgang ebenso schnell außerhalb von Cosmos DB zu erledigen.

Müssen Sie für Ihre jetzt global verteilte Datenbank viele Änderungen an Ihrem Code vornehmen, um zu ermitteln, mit welcher Cosmos DB-Region eine Verbindung hergestellt werden sollte? Wie kann es Ihnen gelingen, diese verschiedenen Versionen Ihrer Anwendungen anhand der Azure-Region, auf der sie laufen, zu pflegen? Die Azure-Plattform nimmt Ihnen diese Arbeit ab!

10.3 Zugriff auf global verteilte Daten

Die Azure-Plattform bestimmt größtenteils den besten Standort für Ihre Anwendung. Anwendungen müssen in der Regel Daten lesen und schreiben können. Sie können für Ihre Cosmos DB-Datenbank Failover-Richtlinien definieren, um die primäre Schreibregion zu kontrollieren. Diese Schreibregion fungiert als zentraler Hub, der eine konsistente Replikation der Daten über die verschiedenen Regionen gewährleistet. Ihre Web App kann allerdings aus mehreren verfügbaren Regionen lesen. Dies führt, zu schnelleren Abfragen und einer schnelleren Rücksendung der Daten an den oder die Kund*in. Diese Abläufe werden alle als REST-Aufrufe behandelt.

Betrachten wir nun, was mit der Azure CLI passiert, wenn Sie Informationen über eine Cosmos-DB-Datenbank abfragen. Dieser Prozess funktioniert wie eine Anwendung, die eine Verbindung zu einer Datenbank herstellt. Aber die CLI hindert Sie daran, zu tief in den Code einzudringen.

Jetzt testen

Mit dem Befehl `az cosmosdb` show erhalten Sie, wie nachfolgend dargestellt, Informationen über Ihre Lese- und Schreibregion.

1 Öffnen Sie das Azure Portal in einem Webbrowser und dann die Cloud Shell.

2 Mit dem Befehl `az cosmosdb show` erhalten Sie Informationen über die Lese- und Schreibregionen für Ihre Cosmos DB-Datenbank.

Geben Sie den Namen der Ressourcengruppe und den Datenbanknamen ein, den Sie in den vorherigen „Jetzt testen”-Übungen erstellt haben. Im folgenden Beispiel heißt die Ressourcengruppe `azuremolchapter10` und der Name der Cosmos DB Datenbank lautet `azuremol`:

```
az cosmosdb show \
  --resource-group azuremolchapter10 \
  --name azuremol
```

Die auf diesen Befehl hin ausgegebenen Informationen sind umfangreich. Wir betrachten die zwei wichtigsten Elemente: Leseregionen und Schreibregionen. Hier ein Beispiel für den Abschnitt `readLocations`:

```
"readLocations": [
    {
      "documentEndpoint":"https://azuremol-eastus.documents.azure.com:443/",
      "failoverPriority": 0,
      "id": "azuremol-eastus",
      "isZoneRedundant": "false",
      "locationName": "East US",
      "provisioningState": "Succeeded"
    },
    {
      "documentEndpoint":
      ➥"https://azuremol-westeurope.documents.azure.com:443/",
      "failoverPriority": 1,
      "id": "azuremol-westeurope",
      "isZoneRedundant": "false",
      "locationName": "West Europe",
      "provisioningState": "Succeeded"
      }
],
```

Beim Aufbau der Verbindung Ihrer Anwendung zu einer Cosmos DB-Datenbank können Sie eine Verbindungsrichtlinie angeben. Arbeiten Sie normalerwiese nicht mit Datenbanken, könnten Sie auf einem Windows-Rechner eine einfache Open Database Connectivity-Verbindung (ODBC) einrichten. Die Verbindungszeichenfolge definiert normalerweise einen Hostnamen, einen Datenbanknamen, einen Port sowie Zugangsdaten. Bei Cosmos DB ist das nicht anders. Sie können aus mehreren Sprachen, wie zum Beispiel .NET, Python, Node.js und Java, eine Verbindung mit einer Cosmos DB herstellen. Unabhängig von der verwendeten Sprache ist bei allen SDKs eine Einstellung gleich: Endpoint Discovery. Zwei Haupteigenschaften der Verbindungsrichtlinie sind bedeutend:

- *Automatische Endpoint Discovery* – das SDK liest alle verfügbaren Endpunkte in der Cosmos DB aus und nutzt die angegebene Failover-Order. Diese Methode gewährleistet, dass Ihre Anwendung stets die von Ihnen auf der Datenbankebene angegebene Reihenfolge einhält. Möglicherweise möchten Sie sämtliche Lesevorgänge über die Region USA Ost abwickeln und Westeuropa nur nutzen, wenn am primären Standort eine Wartung erfolgt.
- *Bevorzugte Standorte für den Endpunkt* – Sie benennen die zu verwendenden Standorte. Das ist beispielsweise der Fall, wenn Sie Ihre App für Westeuropa einsetzen und den Standort als Endpunkt gewährleisten wollen. Ein wenig Flexibilität geht verloren, wenn Sie Endpunkte hinzufügen oder entfernen. Sie stellen damit jedoch sicher, dass sich Ihr Standard-Endpunkt in der Nähe Ihrer App befindet, ohne dass ein umfangreiches Netzwerk-Routing dies für Sie erledigt.

Normalerweise ist dafür das Cosmos DB SDK und nicht Ihre Anwendung zuständig. Ihre Anwendung verändert nichts an der Methode der Datenbankverbindung. Sie weiß nur, das Verbindungen mit verschiedenen Standorten *möglich* sind. Doch das SDK *stellt* die Verbindung her und nutzt diese Standortinformationen.

Abbildung 10.10 zeigt eine vereinfachte Methode zur Nutzung dieser Standortinformationen zwischen Ihrer Anwendung und dem SDK. Auch hier ist die Sprache irrelevant. Die Methode ist dieselbe: Die Abbildung beruht auf Python SDK, da einige Beispiele in dieser Sprache geschrieben wurden. Dieses Beispiel geht auch davon aus, dass Sie automatische Standorte für den Endpunkt nutzen.

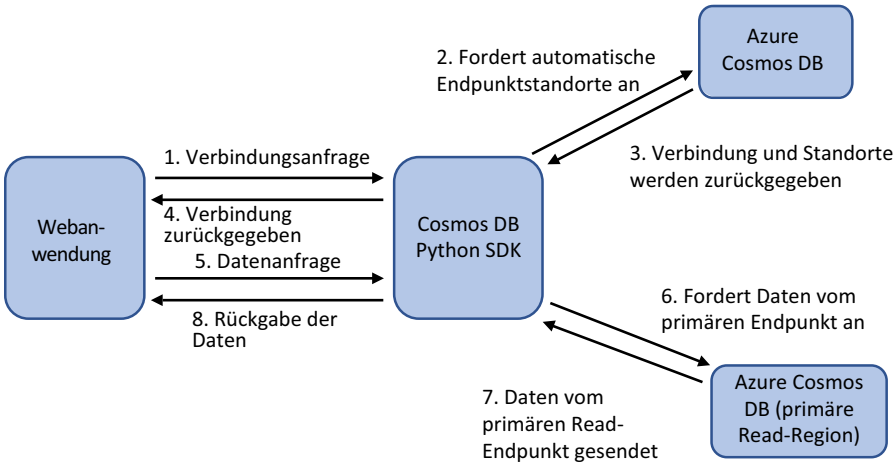


Abbildung 10.10 Ablauf der Anfragen über ein Cosmos DB SDK, wenn eine Anwendung zur Abfrage bei Cosmos DB Standortinformationen verwendet

Abbildung 10.10 verdeutlicht folgende Schritte:

1. Ihre Anwendung benötigt eine Verbindung zu einer Cosmos DB-Datenbank. Aktivieren Sie in der Verbindungsrichtlinie die automatische Endpoint Discovery. Die Anwendung stellt mithilfe des Cosmos DB SDK eine Datenbankverbindung her.
2. Das Cosmos DB SDK stellt eine Verbindungsanfrage und gibt an, dass es automatische Endpunktstandorte nutzen möchte.
3. Eine Verbindung wird anhand der angeforderten Anmeldedaten und der gewünschten Datenbank zurückgegeben.
4. Das SDK gibt ein Verbindungsobjekt für die Anwendung zurück. Die Standortinformationen werden aus der Anwendung abstrahiert.
5. Die Anwendung fragt aus der Cosmos DB-Datenbank einige Daten ab. Das SDK wird erneut verwendet, um Daten abzufragen und zu erhalten.
6. Das SDK nutzt die Liste der verfügbaren Endpunkte und stellt die Anfrage an den ersten verfügbaren Endpunkt. Das SDK verwendet dann den Endpunkt der Verbindung, um die Daten abzufragen. Ist der primäre Endpunkt, etwa aufgrund von Wartungsarbeiten, nicht verfügbar, wird automatisch der nächste Endpunkt verwendet.
7. Cosmos DB gibt die Daten vom Endpunktstandort zurück.

8 Das SDK gibt die Daten von Cosmos DB an die Anwendung zurück, um sie nach Bedarf zu analysieren und anzuzeigen.

Der letzte Punkt in Cosmos DB sind die Zugriffsschlüssel, mit denen Sie steuern können, wer auf die Daten zugreifen kann und welche Berechtigungen dabei gewährt werden.. Es ist möglich, Schlüssel neu zu generieren. Wie bei Kennwörtern ist es ratsam, eine Richtlinie zu implementieren, damit dieser Prozess der Neugenerierung von Schlüsseln regelmäßig erfolgt. Schlüssel sind erforderlich, um auf die verteilten Daten in Cosmos DB zugreifen zu können. Das Azure Portal kann Ihnen alle Schlüssel und Verbindungszeichenfolgen für Ihre Datenbank anzeigen.

Jetzt testen

Gehen Sie folgendermaßen vor, um die Schlüssel für Ihr Cosmos DB-Konto anzuzeigen:

1 Suchen Sie in der Navigationsleiste links im Azure Portal nach „Resource Group" (Ressourcengruppe), und wählen Sie diese Option aus.
2 Wählen Sie die Ressourcengruppe aus, in der Sie Ihre Cosmos DB-Datenbank erstellt haben, wie beispielsweise `azuremolchapter10`.
3 Wählen Sie aus der Liste aller Ressourcen Ihr Cosmos DB-Konto aus.
4 Wählen Sie auf der linken Seite „Keys" (Schlüssel) aus.
5 Achten Sie auf den URI und den Primärschlüssel (siehe Abbildung 10.11). Diese Werte benötigen Sie für die Übung am Ende des Kapitels.

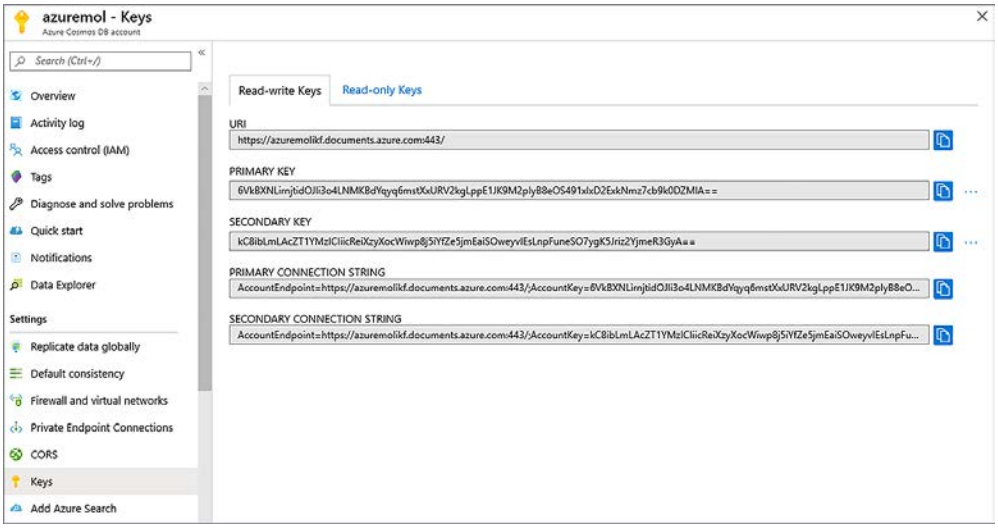


Abbildung 10.11 Im Abschnitt Schlüssel Ihres Cosmos DB-Kontos sind die Verbindungsdaten und die Zugriffsschlüssel aufgelistet. Diese Daten benötigen Sie, wenn Sie, wie in der Übung am Ende des Kapitels, Anwendungen konstruieren und ausführen.

Viele Ereignisse in Cosmos DB geschehen vor dem Hintergrund der Verteilung Ihrer Daten und der Möglichkeit, dass Ihre Anwendungen Daten von den am besten geeigneten Regionen aus lesen und schreiben können. Doch das ist der springende Punkt. Wenn Sie wissen, was der Cosmos DB-Service tut, können Sie Ihre Anwendung besser entwerfen und planen oder Fehler beheben, wenn Anwendungen das SDK nicht wie erforderlich Lese- und Schreibvorgänge durchführen lassen. Über die Methode und den Zeitpunkt brauchen Sie sich keine Gedanken zu machen. Konzentrieren Sie sich auf Ihre Anwendungen und nutzen Sie Azure-Dienste wie Cosmos DB, um die Funktionen und Vorteile der Cloud zu nutzen, die es Ihnen ermöglichen auf globaler Ebene zu agieren.

10.4 Übung: Einsatz einer Web App, die Cosmos DB nutzt

In Abschnitt 10.2.2 haben Sie Ihre Cosmos DB-Datenbank global verteilt. Jetzt haben wir viel Theorie behandelt, zum Beispiel, wie Webanwendungen Daten aus Regionen auf der ganzen Welt lesen können. Jetzt wollen Sie Cosmos DB wahrscheinlich einfach nur in Aktion sehen, also hier ist Ihre Chance! In dieser Übung wird die grundlegende Webanwendung aus den vorherigen Kapiteln verwendet, aber dieses Mal stammt das Pizzamenü aus den Objekten, die Sie in einer vorangegangenen „Jetzt testen“-Übung zur Cosmos DB-Datenbank hinzugefügt haben:

1. Erstellen Sie eine Web App im Azure Portal.
2. Da die Pizzeria keine einfache HTML-Seite mehr ist, wählen Sie `Node LTS` für die Laufzeit, die unter Linux ausgeführt wird.
3. Wenn die Web App bereit ist, erstellen Sie eine Bereitstellungsquelle (lokales Git-Repository). Gehen Sie dabei genauso vor wie in den vorherigen Kapiteln, wie zum Beispiel Kapitel 3. Sehen Sie sich gegebenenfalls diese Übungen noch einmal an.
4. Öffnen Sie die Cloud Shell. In den vorherigen Kapiteln erhielten Sie eine Kopie der Azure-Beispiele von GitHub. Wenn Sie das nicht getan haben, sichern Sie sich eine Kopie wie folgt:

   ```
   git clone https://github.com/fouldsy/azure-mol-samples-2nd-ed.git
   ```

5. Wechseln Sie zu dem Verzeichnis mit dem Beispiel für die Cosmos DB Web App:

   ```
   cd ~/azure-mol-samples-2nd-ed/10/cosmosdbwebapp
   ```

6. Bearbeiten Sie die Konfigurationsdatei mit der Datenbank-URI und dem Zugriffsschlüssel, den Sie in der vorherigen „Jetzt testen"-Übung kopiert haben, um Ihre Cosmos DB-Schlüssel zu sehen:

   ```
   nano config.js
   ```

7. Schreiben Sie die Datei, indem Sie STRG + O drücken, und beenden Sie dann, indem Sie STRG + X drücken.
8. Mit dem folgendem Befehl fügen Sie Ihre Änderungen in Git hinzu und bestätigen sie:

   ```
   git init && git add . && git commit -m "Pizza"
   ```

9 Erstellen Sie in Ihrem Staging-Slot einen Link zu der neuen Git-Bereitstellung mit dem Befehl `git remote add azure`, gefolgt von Ihrer Git-Bereitstellungs-URL.

10 Nutzen Sie den Befehl `git push azure master`, um Ihre Änderungen an Ihre Web App zu übertragen.

11 Wählen Sie aus dem Übersichtsfenster im Azure Portal die URL Ihrer Web App.

12 Öffnen Sie diese URL in einem Webbrowser, um Ihre Pizzeria anzuzeigen, die jetzt auf Cosmos DB betrieben wird, wie in Abbildung 10.12 dargestellt.

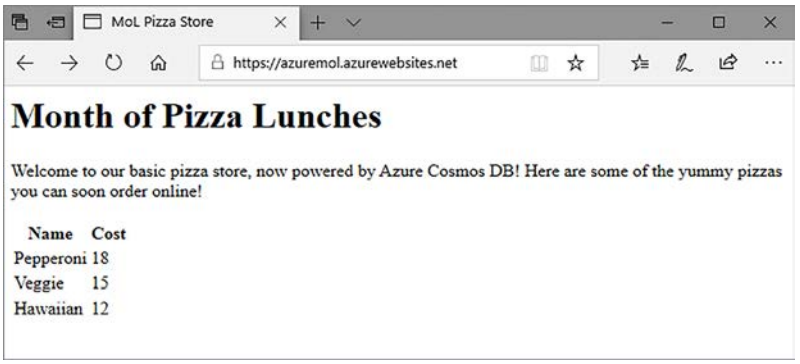


Abbildung 10.12 Die einfache Azure Web App zeigt Ihre kurze, anhand der Daten in der Cosmos DB-Datenbank erstellte Speisekarte. Die Pizzeria aus früheren Kapiteln wird gezeigt, aber jetzt wird die Liste der Pizzen und deren Preise von Cosmos DB unterstützt. Die Website ist weiterhin einfach, da das Ziel darin besteht, den Dienst in der Praxis zu sehen und zu verstehen, wie Sie mit der Entwicklung ihrer eigenen Anwendungen beginnen können.

11 Verwalten von Netzwerkverkehr und Routing

Die DNS-Auflösung (Domain Name System) ist das Herzstück fast jeder digitalen Verbindung, die Sie herstellen. Sie ist relevant, wenn Sie im Internet surfen, E-Mails empfangen, Netflix sehen oder Skype-Anrufe tätigen. DNS übersetzt einen Namen, wie manning.com, in eine IP-Adresse. Wer sich mit einem neuen Thema befassen möchte, muss sich nicht 35.166.24.88 merken, sondern braucht nur manning.com in einen Webbrowser einzugeben und kann sich über das Angebot informieren! Netzwerkgeräte leiten den Datenverkehr anhand von IP-Adressen weiter. Es bedarf also einer Methode, die es leichter macht, sich Adressen zu merken, um Dinge zu tun, wie online Bücher zu kaufen oder eine Pizza zu bestellen.

In den vorherigen Kapiteln haben wir uns ausführlich mit der Entwicklung von Anwendungen befasst, die eine Skalierung, hohe Verfügbarkeit und globale Verteilung ermöglichen. Was nun noch fehlt, ist unter anderem eine Methode, um Kunden auf der ganzen Welt auf die am besten geeignete Anwendungsinstanz zu leiten, sprich die Instanz, die ihnen am nächsten ist. Mithilfe des Azure Traffic Managers leiten Sie Ihre Kund*innen ganz einfach und automatisch anhand von Kriterien wie Leistung oder geografische Region auf Ihre Anwendungsinstanz um. In diesem Kapitel erfahren Sie, wie Sie DNS-Zonen auf Azure erstellen und verwalten können. Wir erklären Ihnen, wie Sie den Traffic Manager nutzen können, um Kund*innen mit DNS-Abfragen umzuleiten (siehe Abbildung 11.1).

11.1 Was ist Azure DNS?

Sie benötigen keine weiterführendenden Kenntnisse der Funktionsweise von DNS, um dieses Kapitel zu lesen und Azure DNS zu verwenden. Abbildung 11.2 gibt Ihnen einen ausführlichen Überblick darüber, wie ein oder eine Benutzer*in einen DNS-Dienst abfragt, um die IP-Adresse für eine Webanwendung zu erhalten. Zwischen den Schritten 1 und 2 kann es viele Zwischenschritte geben. Wenn Sie in Ihrer Mittagspause am Ende dieses Kapitels noch etwas Zeit haben, dann lesen Sie doch, wie DNS-Abfragen und rekursive Anfragen funktionieren.

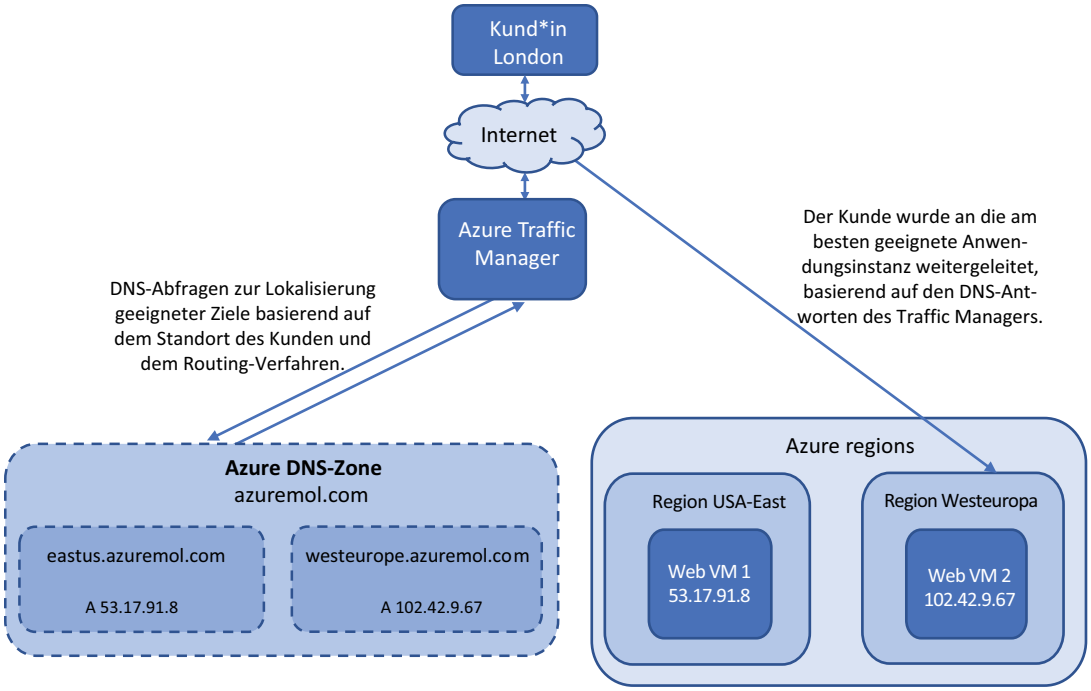


Abbildung 11.1 **In diesem Kapitel untersuchen wir, wie Sie DNS-Zonen mit Azure DNS erstellen können. Der Traffic Manager kann für DNS-Abfragen und zur Weiterleitung der Kund*innen auf die nächstgelegene Anwendungsinstanz verwendet werden. Dies führt zu einer geringeren Latenz und besseren Reaktionszeiten.**

Azure DNS funktioniert wie jede andere DNS-Lösung, die Sie möglicherweise kennen. Ihre Zone und Ihre Datensätze werden in Azure gespeichert. Die Namensserver, die auf DNS-Abfragen reagieren, sind global über die Azure-Rechenzentren verteilt.

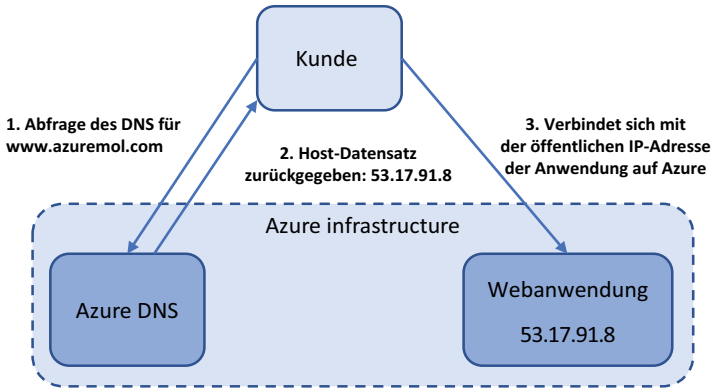


Abbildung 11.2 **Dieser vereinfachte DNS-Datenfluss verdeutlicht, wie ein Benutzender eine DNS-Abfrage für www.azuremol.com an einen DNS-Server sendet, eine Antwort inklusive der zugehörigen IP-Adresse erhält, und sich dann mit der Webanwendung verbinden kann.**

Azure DNS unterstützt alle Datensatztypen, die Sie von einem guten DNS-Serviceangebot erwarten würden. Es können sowohl IPv4- als auch IPv6-Einträge erstellt werden. Folgende Datensatztypen werden unterstützt:

- *A* – IPv4-Host-Einträge, die dazu dienen, Kund*innen auf Anwendungen und Services zu verweisen
- *AAAA* – IPv6-Host-Einträge für die Anwender*innen, die IPv6 nutzen, um Kund*innen auf Ihre Anwendungen und Dienste zu verweisen
- *CNAME* – Canonical Name oder Alias, Einträge, die zum Beispiel einen Kurznamen darstellen, der einfacher zu verwenden ist, als der volle Hostname eines Servers
- *MX* – Mail Exchange-Einträge, die E-Mail-Verkehr an Ihre Mail-Server oder Ihren Mail-Provider leiten
- *NS* – Nameserver-Einträge, die automatisch generierte Datensätze für die Azure-Nameserver enthalten
- *PTR* – Pointer Resource Records für umgekehrte DNS-Abfragen, die dazu dienen, IP-Adressen auf Host-Namen abzubilden
- *SOA* – Start of Authority-Einträge, die automatisch generierte Einträge für die Azure-Nameserver enthalten
- *SRV* – Service Resource Records zur Erkennung von Netzwerkdiensten, etwa zur Identitätsfindung
- *TXT* – Text Resource Records, wie zum Beispiel für Sender Protection Framework (SPF) oder DomainKeys Identified Mail (DKIM)

In einer typischen DNS-Konfiguration können Sie mehrere DNS-Server konfigurieren. Selbst bei der geografischen Verteilung dieser Server zu Redundanz-Zwecken kann es vorkommen, dass Kund*innen einen Nameserver auf der anderen Seite der Welt abfragen. Auch wenn eine DNS-Abfrage, Auflösung und Beantwortung nur Millisekunden dauert, kann sich das erheblich auf die benötigte Zeit auswirken, wenn gerade sehr viele Kund*innen eine Pizza bestellen möchten.

Azure-DNS-Zonen werden weltweit über die Azure-Rechenzentren repliziert. Die *Anycast* Netzwerkadressierung gewährleistet, dass bei einer DNS-Abfrage Ihrer Domain der dem oder der Kund*in nächstgelegene Nameserver die Anfrage beantwortet. Wie erfolgt die Adressierung mit Anycast? In der Regel wird eine einzige IP-Adresse über mehrere Regionen hinweg beworben. Anstatt eine einfache DNS-Abfrage zu bearbeiten, die sich auf eine einzige IP-Adresse an einem einzigen Ort bezieht, definiert die Anycast Adressierung intelligent, woher eine Anfrage kommt, und leitet den oder die Kund*in auf die nächstgelegene beworbene Region. Durch dieses Routing können sich Ihre Kund*innen schneller mit Ihrer Webanwendung verbinden. Dies verbessert das allgemeine Kundenerlebnis.

Sie müssen kein*e Netzwerkexpert*in sein, um zu verstehen, wie das funktioniert. Azure nimmt Ihnen die Arbeit ab! In Abschnitt 11.2 geht es um die Kombination von Azure DNS mit dem Azure Traffic Manager. Damit werden DNS-Abfragen von den nächstgelegenen Nameservern beantwortet, und Ihre Kund*innen werden mit der nächstgelegenen Anwendungsinstanz verbunden. Das sind Millisekunden, auf die es ankommt!

11.2 Übertragen einer echten Domain auf Azure DNS

Wenn Sie eine echte Domäne registrieren, erhalten Sie von Ihrem Provider eine Verwaltungsschnittstelle und Tools zum Verwalten dieser Domäne. Damit Kund*innen auf Ihre Dienste zugreifen und die Azure DNS-Zone und die Datensätze nutzen können, delegieren Sie die Zuständigkeit für Ihre Domäne an die Azure Nameserver.

Diese Delegation führt dazu, dass alle DNS-Abfragen sofort an die entsprechenden Azure-Nameserver weitergeleitet werden (siehe Abbildung 11.3). Über die Plattform von Azure können Sie derzeit keine Domänen erwerben und registrieren. Folglich müssen Sie den Domänennamen über eine externe Registrierungsstelle erwerben und können dann die NS-Einträge auf die Azure-Nameserver verweisen.

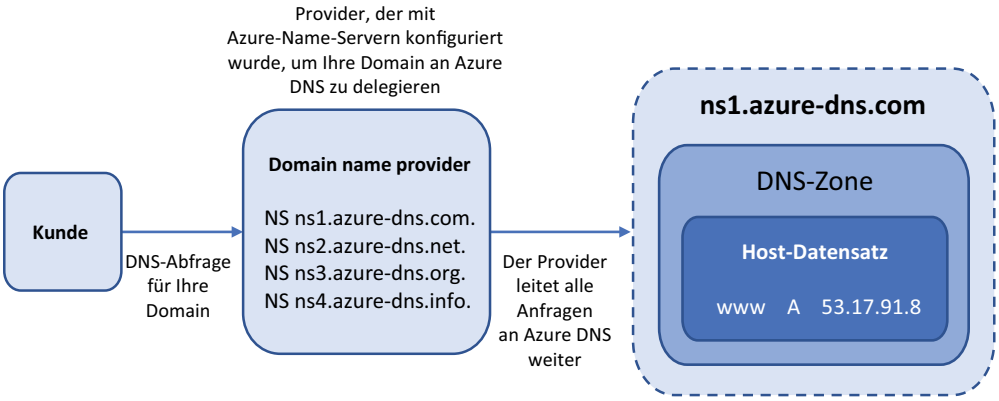


Abbildung 11.3 Um Ihre Domäne an Azure zu delegieren, konfigurieren Sie diese bei Ihrem aktuellen Domain-Provider mit Ihren Azure Nameserver-Adressen. DNS-Abfragen von Kund*innen für Ihre Domäne werden direkt an die Azure-Nameserver Ihrer Zone gesendet.

Warum sollten Sie Ihre DNS an Azure delegieren? Sie vereinfachen die Verwaltung und den Betrieb. Azure bietet eine einzigartige Verwaltungsschnittstelle, über die Sie Aufgaben wie zusätzliche Services, die Konfiguration der Load Balancer oder verbesserte Reaktionszeiten mit global replizierter DNS bewältigen können. Wenn Sie Ihre DNS-Zonen auf Azure hosten, können Sie einige der in Kapitel 6 behandelten Sicherheitsmerkmale des Ressourcenmanagers implementieren. Nutzen Sie Funktionen wie die rollenbasierte Zugriffskontrolle (RBAC), mit der Sie den Zugang zu den DNS-Zonen begrenzen und überprüfen können, oder Ressourcensperren, mit denen Sie ein unbeabsichtigtes oder gar böswilliges Löschen von Zonen vermeiden können.

Die meisten Domänenregistrierungsstellen bieten Ihren Kund*innen einfache Benutzungsoberflächen und Steuerelemente für die Verwaltung von DNS-Zonen und -Einträge. Azure DNS reduziert den Verwaltungsaufwand und erhöht die Sicherheit. Sie können Azure CLI, Azure PowerShell oder die REST APIs nutzen, um Datensätze hinzufügen oder zu bearbeiten. Operative Abteilungen können die gleichen Tools und Workflows nutzen, um neue Dienstleistungen einzuführen. Eine Fehlerbehebung ist bei Problemen oft einfacher, wenn Sie sicherstellen können, dass das DNS erwartungsgemäß funktioniert, ohne die Variable eines weiteren DNS-Providers einführen zu müssen.

Wenn Sie es für sinnvoll halten, Ihre Domäne auf Azure DNS zu delegieren, stellt sich die Frage: Auf welche Azure-Nameserver sollte Ihre Domäne verweisen? Beim Erstellen einer Azure DNS-Zone werden Ihnen die Nameserver im Portal angezeigt (siehe Abbildung 11.4). Sie können auch mit Azure CLI oder Azure PowerShell auf diese Nameserver-Adresse zugreifen.

Zu diesem Thema gab es keine „Jetzt testen"-Übungen auf den vorherigen Seiten, denn Sie können die Weiterleitung echten

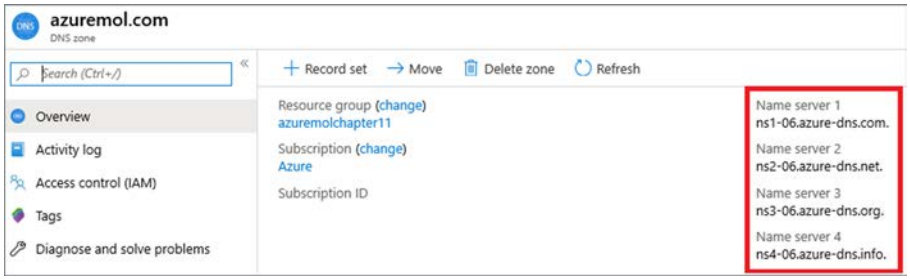


Abbildung 11.4 Die Azure-Nameserver für Ihre DNS-Zone finden Sie im Azure Portal, in der Azure CLI oder der Azure PowerShell.

Datenverkehrs erst testen, wenn Sie eine echte Domäne kaufen und konfigurieren. Sie können eine Azure DNS-Zone ohne eine echte Domäne erstellen, aber dann ist es nicht möglich, Datenverkehr darauf zu leiten. In der Praxis aktualisieren Sie die NS-Datensätze mit Ihrem momentanen Provider derart, dass alle Abfragen für Ihre Domäne auf die Azure-Nameserver verweisen. Es kann 24 bis 48 Stunden (in der Regel viel weniger Zeit) dauern, bis Ihre Domäne delegiert und in der gesamten globalen DNS-Hierarchie vorhanden ist. Berücksichtigen Sie das bei Ihrer Planung, es kann für Ihre Kund*innen beim Zugriff auf Ihre Anwendung zu kurzen Unterbrechungen kommen.

11.3 Globales Routing und Auflösung mit Traffic Manager

In früheren Kapiteln haben wir hochverfügbare, weltweit verteilte Anwendungen behandelt. Letztendlich geht es darum, mehrere Web App- oder VM-Instanzen in verschiedenen Regionen oder Kontinenten zu etablieren, die sich mit einer Cosmos DB-Instanz in der Nähe Ihrer Kund*innen verbinden. Doch wie kann es Ihnen gelingen, Ihre*innen Kunden mit der nächstgelegenen VM oder Web App zu verbinden, die Ihre Anwendung betreibt?

Azure Traffic Manager ist ein Netzwerk-Service, der als zentrales Ziel für Ihre Kund*innen fungiert. Nehmen wir das Beispiel einer Webanwendung unter der Adresse www.azuremol.com. Abbildung 11.5 gibt einen Überblick darüber, wie Traffic Manager die Nutzer*innen auf die nächstgelegene verfügbare Anwendung weiterleitet.

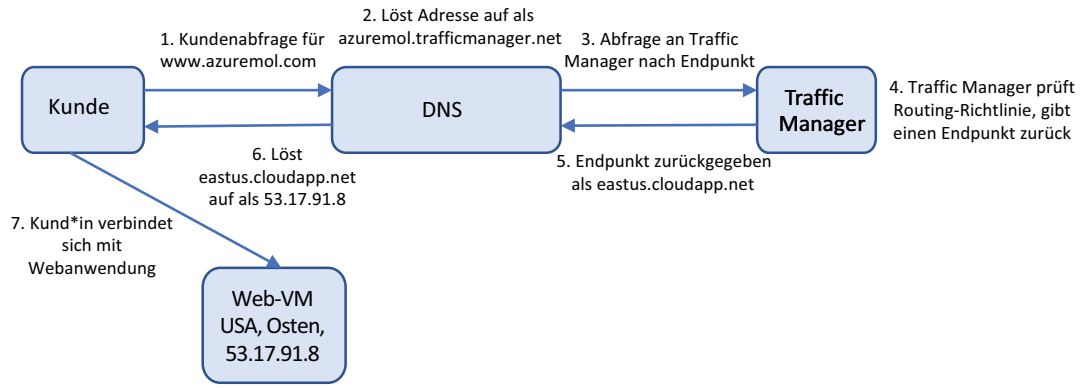


Abbildung 11.5 Ein*e Kund*in schickt eine DNS-Abfrage an einen DNS-Dienst für www.azuremol.com. Der DNS-Dienst leitet die Abfrage an den Traffic Manager weiter, der anhand der verwendeten Routing-Methode einen Bereitstellungpunkt zurückgibt. Der Endpunkt wird mit einer IP-Adresse aufgelöst, über die der Kunde eine Verbindung zur Webanwendung herstellt.

Traffic Manager übernimmt nicht die Rolle eines Load Balancers, um den es in Kapitel 8 ging. Der Traffic Manager leitet Datenverkehr an eine öffentliche IP weiter. Betrachten wir den Datenverkehrsfluss etwas genauer:

1. Der oder die Benutzer*in stellt eine DNS-Abfrage für www.azuremol.com. Sein DNS-Server kontaktiert die Nameserver für azuremol.com und erfragt den Datensatz für www. Sollten Sie Azure DNS verwenden, könnte es sich dabei um Azure Nameserver handeln.
2. Der www-Host löst sich in einen CNAME Resource Record auf, der auf azuremol.trafficmanager.net verweist.
3. Der DNS-Dienst leitet die DNS-Abfrage an die Azure-Namenserver für trafficmanager.net weiter.
4. Traffic Manger prüft anschließend die Abfrage und bestimmt einen Bereitstellungpunkt, auf den der oder die Benutzer*in geleitet wird. Zustand und Status des Bereitstellungpunktes werden, wie die Azure Load Balancer, analysiert. Auch die Routing-Methode des Traffic Managers wird überprüft. Traffic Manager kann folgende Routing-Methoden verwenden:
   - *Priorität* – kontrolliert die Reihenfolge, in der Bereitstellungpunkte abgerufen werden
   - *Gewichtet* – verteilt den Datenverkehr anhand einer zugewiesenen Gewichtung an die Endpunkte
   - *Leistung* – latenzbasierte Weiterleitung der Benutzer*innen auf einen Endpunkt, damit dieser die schnellstmögliche Reaktionszeit erhält
   - *Geografisch* – assoziiert Endpunkte mit einer geografischen Region und leitet die Benutzer*innen anhand ihres Standortes dorthin
5. Der Endpunkt eastus.cloudapp.net wird von Traffic Manager an den DNS-Dienst zurückgegeben.
6. Der DNS-Dienst ermittelt den DNS-Eintrag für eastus.cloudapp.net und gibt das Ergebnis der Abfrage an den oder die Kund*in zurück.
7. Mit der IP-Adresse des gewünschten Endpunktes kontaktiert der oder die Kund*in die Webanwendung direkt. An dieser Stelle könnte der Datenverkehr eher die öffentliche IP-Adresse eines Azure Load Balancers als direkt eine VM erfassen.

Die Aufgabe des Traffic Managers besteht darin, einen Bereitstellungpunkt einer Anwendung zu bestimmen und Kund*innen darauf zu leiten. Einige Integritätsprüfungen überwachen den Status der Endpunkte. Sie funktionieren ähnlich wie die in Kapitel 8 beschriebenen Lastenausgleichsmodule. Sie können eine Priorität oder ein gewichtetes Traffic-Routing definieren, um Benutzer*innen über eine Reihe verfügbarer Endpunkte zu verteilen. Auch dies ist mit dem Load Balancer vergleichbar. Der Traffic Manager leitet den Datenverkehr in der Regel auf einen Azure Load Balancer oder Anwendungsgateway oder an eine Web App weiter.

Azure Front Door

Der Traffic Manager, den wir in diesem Abschnitt betrachten, eignet sich hervorragend für die globale Verteilung und das Routing von Datenverkehr. Es funktioniert mit allen Arten von Internetendpunkten, nicht nur mit Ressourcen in Azure. Das Routing von Datenverkehr basiert auf DNS und betrachtet nicht die eigentliche Anwendung selbst.

Wenn Sie Datenverkehr-Routing auf Anwendungsebene und die Möglichkeit zur Durchführung von TLS/SSL-Offloading oder Pro-HTTP/HTTPS-Anforderungsweiterleitung benötigen, hilft Ihnen Azure Front Door. Traffic Manager und Front Door bieten die gleiche

(Fortsetzung)

Art von Dienst- und Konfigurationsoptionen, aber Front Door ist insbesondere für die Arbeit auf der Anwendungsebene konzipiert. Front Door hat auch einige coole Performance-Tricks, wie z. B. ein geteiltes TCP, um Verbindungen in kleinere Stücke zu zerlegen und die Latenz zu reduzieren.

In Kapitel 8 haben wir uns mit Load Balancer und dem erwähnten Anwendungsgateway befasst, das auf der Anwendungsebene funktioniert und beispielsweise TLS-Verschiebungen ausführt. Der Schwerpunkt in diesem Kapitel lag auf Load Balancern, die Ihnen dabei helfen, die Kernkonzepte zu erlernen, auf denen das Anwendungsgateway aufbauen würde. Dasselbe gilt für hier. Wir konzentrieren uns auf Traffic Manager in diesem Kapitel, obwohl viele der gleichen Konzepte und Konfigurationsoptionen, wie z. B. Routingoptionen, auch für Azure Front Door verfügbar sind. Wie bei den meisten Dingen in Azure wird das, was in jedem Dienst verwendet wird, von den ausgeführten Anwendungen und Ihren Anforderungen gesteuert.

11.3.1 Erstellen von Traffic Manager-Profilen

Traffic Manager bestimmt mithilfe von Profilen die zu verwendende Routing-Methode sowie die, mit für eine bestimmte Abfrage assoziierten, Endpunkte. Ganz im Sinne des Themas der vorherigen Kapitel über eine global verteilte Anwendung ist es in Ihrem Interesse, dass Ihre Benutzer*innen die ihnen nächstgelegene Webanwendung nutzen. Es gibt dafür zwei verschiedene Routing-Methoden:

- *Routing nach Leistung* – der oder die Kund*in wird auf den Endpunkt mit der, in Bezug auf die Quelle, geringsten Latenz weitergeleitet. Diese intelligente Routing-Methode ermöglicht dem Traffic Manager die Weiterleitung des oder der Kund*in an einen verfügbaren Endpunkt.
- *Geografisches Routing* – der oder die Kund*in wird anhand der Quelle seiner Abfrage immer zu einem bestimmten Endpunkt weitergeleitet. Kund*innen in den USA ist werden beispielsweise immer nach USA Ost weitergeleitet. Bei dieser Routing-Methode müssen Sie zu jedem Bereitstellungpunkt geografische Regionen definieren.

Beim geografische Routing erhalten Sie eine etwas größere Kontrolle über die von den Kund*innen genutzten Endpunkte. Es kann aus regulatorischen Gründen sein, dass staatliche Kund*innen in einer bestimmten Region immer die Bereitstellungpunkte in derselben Region nutzen müssen. Bei den Übungen werden geografische Bereitstellungpunkte genutzt, um einen praxisbezogenen Ansatz zu verfolgen, denn beim geografischen Routing gibt es einen Trick: Sie sollten ein *untergeordnetes Profil* angeben und keinen direkten Endpunkt.

Die Welt wird nicht gleich untergehen, wenn Sie geografisches Routing mit Endpunkten verwenden, wir empfehlen aber, den Datenverkehr auf den letztendlichen Endpunkt mit einem anderen Traffic Manager-Profil zu übergeben. Warum? Regionen können nur mit einem Traffic Manager-Profil assoziiert werden. In den vorherigen Kapiteln über hohe Verfügbarkeit sollten Sie stets für Redundanz sorgen. Wenn Sie eine Region mit einem bestimmten Endpunkt assoziieren und geografisches Routing nutzen, gibt es bei Problemen mit diesem Bereitstellungspunk oder bei Wartungsarbeiten keine Failover-Option.

Stattdessen können Sie verschachtelte untergeordnete Profile nutzen, um Prioritäten zu definieren, die den Datenverkehr immer an einen betriebsbereiten

Bereitstellungpunkt leiten. Ist der Endpunkt fehlerhaft, wird der Datenverkehr an einen alternativen Bereitstellungpunkt geleitet. Abbildung 11.6 veranschaulicht Datenverkehr, der auf eine andere Region überspringt, obwohl es auch möglich wäre, mehrere Web App-Instanzen in USA West zu erstellen und in dem untergeordneten Profil eine gewichtete Routing-Methode zu verwenden. Lassen Sie sich Zeit, wenn Sie den Umfang Ihrer Anwendungsumgebung bestimmen, und finden Sie heraus, wie Sie hinter dem Traffic Manager am besten hochverfügbare Endpunkte einrichten können. Erstellen Sie für diese Beispiele ein Failover zwischen den Regionen, damit Verhaltensunterschiede deutlich werden.

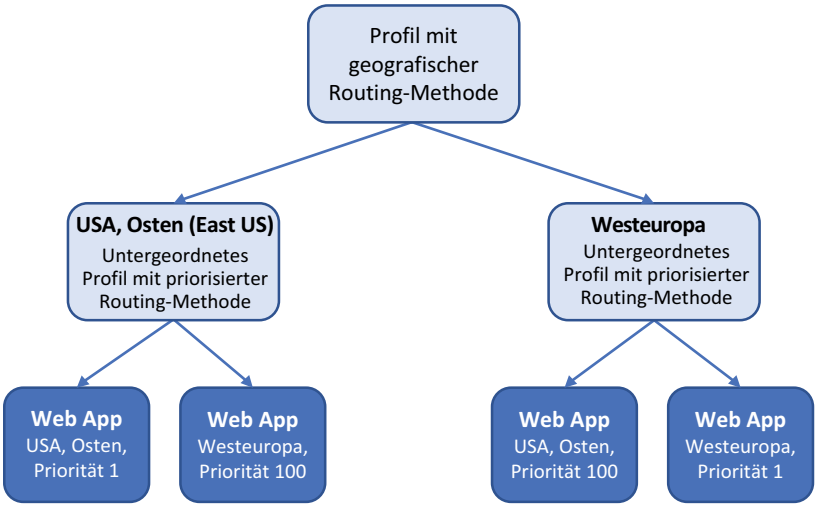


Abbildung 11.6 Ein übergeordnetes Traffic Manager-Profil mit geografischem Routing sollte untergeordnete Profile mit mehreren Endpunkten enthalten. Diese untergeordneten Endpunkte können dank des Prioritätenroutings den Datenverkehr auf den bevorzugten Endpunkt lenken. So sendet das untergeordnete Profil für USA Ost den Datenverkehr immer an den Endpunkt in USA Ost, sofern dieser fehlerfrei läuft. Ist der Endpunkt fehlerhaft, wird der Datenverkehr nach Westeuropa geleitet. Ohne dieses untergeordnete Profil gäbe es für Kund*innen in der Region USA Ost keinen Failover an einen alternativen Endpunkt, und sie könnten nicht auf Ihre Webanwendung zugreifen.

Jetzt testen

Gehen Sie folgendermaßen vor, um Traffic Manager-Profile für Ihre Anwendung zu erstellen.

Bei den restlichen Übungen geht es um USA Ost und Westeuropa. Wenn Sie nicht in einer dieser Regionen leben, wählen Sie eine andere, passendere Region aus. Sie sollten allerdings bei allen Übungen konsequent vorgehen! Die Übung am Ende dieses Kapitels verdeutlicht die Zusammenhänge und Funktionsweise. Wenn Sie außerhalb von Nordamerika oder Europa leben und die Region nicht entsprechend ändern, werden Sie allerdings nicht korrekt auf Ihre Web Apps geleitet werden.

1. Öffnen Sie das Azure Portal, und klicken Sie oben im Dashboard auf das Cloud Shell-Symbol.
2. Erstellen Sie eine Ressourcengruppe und geben Sie einen Namen, wie `azuremolchapter11`, und einen Standort, wie `eastus`, ein:

```
az group create --name azuremolchapter11 --location eastus
```

3 Erstellen Sie ein übergeordnetes Traffic Manager-Profil. Wählen Sie geografisches Routing als Methode und geben Sie einen Namen ein, wie `azuremol`. Der Parameter für den DNS-Namen muss eindeutig sein, geben Sie daher einen eindeutigen Namen ein. Die folgende Domäne erstellt den Hostnamen azuremol.trafficmanager.net, mit dem Sie die Web Apps in der Übung am Ende des Kapitels konfigurieren:

```
az network traffic-manager profile create \
  --resource-group azuremolchapter11 \
  --name azuremol \
  --routing-method geographic \
  --unique-dns-name azuremol
```

4 Erstellen Sie ein untergeordnetes Traffic Manager-Profil. Nutzen Sie dieses Mal das Routing nach Priorität und den Namen eastus. Geben Sie einen weiteren eindeutigen DNS-Namen ein, wie zum Beispiel `azuremoleastus`:

```
az network traffic-manager profile create \
  --resource-group azuremolchapter11 \
  --name eastus \
  --routing-method priority \
  --unique-dns-name azuremoleastus
```

5 Erstellen Sie ein weiteres untergeordnetes Traffic Manager-Profil mit dem Namen westeurope sowie einen weiteren eindeutigen DNS-Namen, wie `azuremolwesteurope`:

```
az network traffic-manager profile create \
  --resource-group azuremolchapter11 \
  --name westeurope \
  --routing-method priority \
  --unique-dns-name azuremolwesteurope
```

6 Nachdem Sie bereits mehrere Male eine Web App entwickelt haben, verwenden wir nun die CLI, um zwei App-Service-Pläne sowie anschließend für jeden Plan eine Web App zu erstellen. Eine dieser Web Apps befindet sich in USA Ost, die andere in Westeuropa. In der Übung am Ende des Kapitels laden Sie Beispiel-Websites für diese Web Apps hoch. Erstellen Sie einstweilen eine leere Website, und richten Sie ein einsatzbereites lokales Git-Repository ein.

Erstellen Sie die Web App in USA Ost folgendermaßen:

```
az appservice plan create \
  --resource-group azuremolchapter11 \
  --name appserviceeastus \
  --location eastus \
  --sku S1
az webapp create \
  --resource-group azuremolchapter11 \
  --name azuremoleastus \
  --plan appserviceeastus \
  --deployment-local-git
```

7 Erstellen Sie eine zweite Web App in der Region Westeuropa:

```
az appservice plan create \
  --resource-group azuremolchapter11 \
```

```
    --name appservicewesteurope \
    --location westeurope \
    --sku S1
  az webapp create \
    --resource-group azuremolchapter11 \
    --name azuremolwesteurope \
    --plan appservicewesteurope \
    --deployment-local-git
```

11.3.2 Globale Verteilung des Datenverkehrs auf die nächste Instanz

Sie haben Traffic Manager-Profile und Endpunkte erstellt, aber keinen Datenverkehr. Würden Kund*innen auf die Profile geleitet, gäbe es keine Verbindung zu Ihren Endpunkten. Das Diagramm in Abbildung 11.7 zeigt Ihnen, wie Sie Endpunkte mit Profilen verknüpfen.

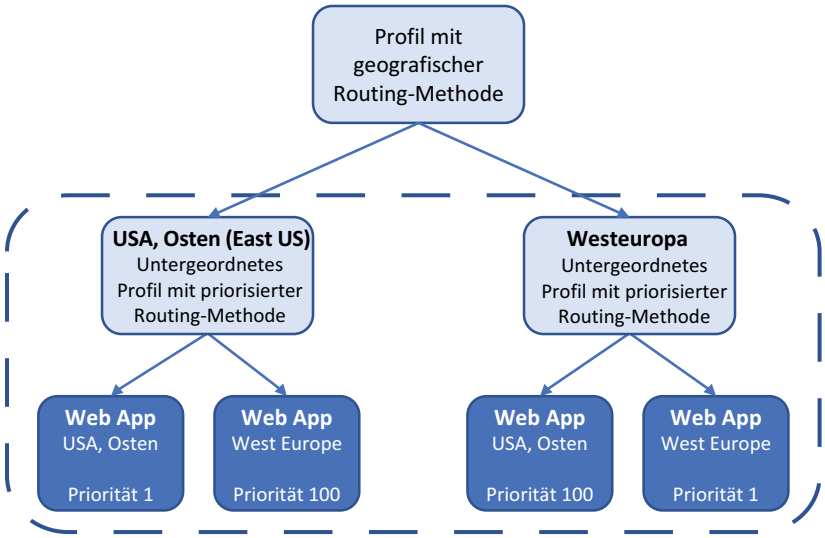


Abbildung 11.7 In diesem Abschnitt verbinden Sie Ihre Endpunkte mit den Traffic Manager-Profilen und definieren die Priorität für den zu verteilenden Datenverkehr.

Ihre ersten von Ihnen geknüpften Verbindungen sind für die Endpunkte Ihrer Web App gedacht. Sie möchten eine hohe Verfügbarkeit gewährleisten. Daher sollten beide Web Apps in jedem Traffic Manager-Profil vorhanden sein. Sie verwenden das Routing nach Priorität, um den gesamten Datenverkehr auf die primäre Web App für jedes Profil zu lenken. Ist diese Web App nicht verfügbar, kann ein Failover des Datenverkehrs auf den Endpunkt der sekundären Web App erfolgen.

Als Sie im Abschnitt 11.3.1 die Traffic Manager-Profile erstellt haben, kamen bei der Integritätsprüfung und der Überwachung der Endpunkte einige Standardeinstellungen als Optionen zum Einsatz. Diese Optionen sind:

- *DNS Time to Live (TTL): 30 Sekunden* – legt fest, wie lange der Traffic Manager die DNS-Antworten zwischenspeichern kann. Eine kurze TTL gewährleistet, dass der Kundendatenverkehr bei der Aktualisierung der Traffic Manager-Konfiguration adäquat gelenkt wird.

- *Endpunkt-Überwachungsprotokoll: HTTP* – Sie können auch HTTPS oder eine einfache TCP-Prüfung wählen. Wie die Load Balancer gewährleisten auch HTTP oder HTTPS, dass jeder Bereitstellungpunkt HTTP 200 OK als Antwort zurückgibt.
- *Port: 80* – der an jedem Endpunkt zu überprüfende Port.
- *Pfad:/* – überprüft standardmäßig das Stammverzeichnis des Endpunktes. Es ist auch möglich, eine benutzerdefinierte Seite zu konfigurieren, wie zum Beispiel eine von Load Balancern verwendete Seite mit Integritätsprüfung.
- *Endpunkt-Testintervall: 30 Sekunden* – Häufigkeit der Überprüfung des Endpunktes. Mögliche Werte sind 10 oder 30 Sekunden. Bei einem schnellen Test alle zehn Sekunden fallen zusätzliche Gebühren pro Endpunkt an.
- *Anzahl tolerierter Fehler: 3* – gibt an, wie oft die Integritätsprüfung eines Endpunktes fehlschlagen kann, bevor er als nicht verfügbar markiert wird.
- *Zeitüberschreitung Test: zehn Sekunden* – Zeitdauer, bis ein Test als fehlgeschlagen markiert und der Endpunkt erneut getestet wird.

Sie brauchen diese Standardoptionen nicht zu verändern. Wenn Sie im echten Einsatz Ihre eigene Anwendungsumgebung erstellen, können Sie bei kritischen Workloads die Anzahl tolerierbarer Fehler oder das Testintervall herabsetzen. Dadurch gewährleisten Sie, dass Integritätsprobleme schnell erkannt werden und der Datenverkehr früher auf einen anderen Endpunkt geleitet wird.

Jetzt testen

Führen Sie folgende Schritte aus, um Endpunkte mit Profilen zu verknüpfen und das geografische Routing abzuschließen:

1. Gehen Sie im Azure Portal auf Ressourcengruppe. Wählen Sie für diese Übung das Traffic Manager-Profil aus, das Sie für USA Osten ausgewählt haben.
2. Gehen Sie in der Navigationsleiste links im Profil auf Endpunkte und anschließend auf „Add“ (Hinzufügen).
3. Erstellen Sie einen Azure-Endpunkt, und geben Sie einen Namen ein, z. B. `eastus`.
4. Es gibt verschiedene Zielressourcentypen. Verwenden Sie App Service. Wählen Sie für die Zielressource Ihre Web App in USA Ost, wie z. B. azuremoleastus.
5. Lassen Sie die Priorität auf 1, akzeptieren Sie alle anderen festgelegten Standardwerte, und wählen Sie dann OK.
6. Wiederholen Sie den Vorgang, und fügen Sie einen weiteren Endpunkt hinzu. Nennen Sie diesen Endpunkt `westeurope`. Wählen Sie Ihre Web App in Westeuropa als Zielressource, und setzen Sie die Priorität auf 100.

   Ihr Traffic Manager-Profil listet nun zwei Endpunkte auf: Einen für die Web App in USA Ost und einen für die Web App in Westeuropa (siehe Abbildung 11.8). Dieses Routing nach Priorität der Endpunkte leitet den Datenverkehr immer auf die Web App in USA Ost, wenn diese Ressource fehlerfrei läuft. Ist diese Ressource nicht verfügbar, wird zur Entlastung die Web App in Westeuropa angesteuert.

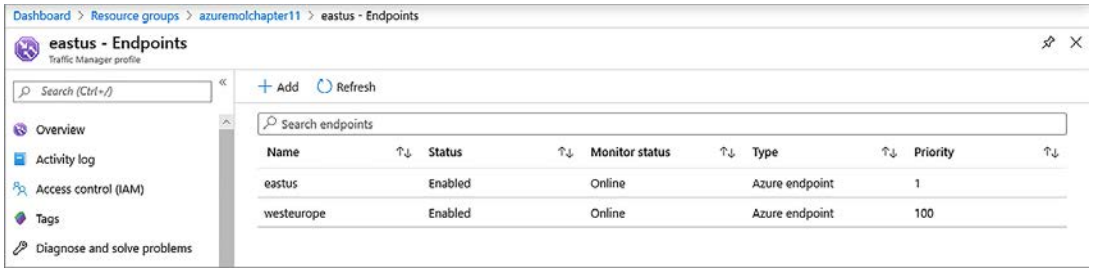


Abbildung 11.8 Im Traffic Manager-Profil sind zwei Endpunkte aufgeführt. Der Endpunkt für USA Osten hat geringere Priorität. Der Datenverkehr läuft also immer dann darüber, wenn der Endpunkt fehlerfrei läuft. Redundanz erfolgt über den Endpunkt Westeuropa. Er wird nur genutzt, wenn der Endpunkt USA Osten nicht verfügbar ist.

7. Gehen Sie zu Ihrer Ressourcengruppe, und wählen Sie das Traffic Manager-Profil für Westeuropa aus.
8. Wählen Sie Endpunkte hinzufügen.
9. Wiederholen Sie die Schritte zum Hinzufügen von zwei Endpunkten, und konfigurieren Sie sie wie folgt:
   - Name: westeurope
     Zielressource: Web App in Westeuropa
     Priorität: 1
   - Name: eastus
     Zielressource: Web App in USA Osten
     Priorität: 100

Ihr Traffic Manager-Profil listet nun zwei Endpunkte auf: Einen für die Web App in Westeuropa und einen für die Web App in USA Osten (siehe Abbildung 11.9). Sie stellen damit die gleiche Redundanz wie bei dem vorherigen Traffic Manager-Profil bereit. Dieses Mal geht der gesamte Datenverkehr bei fehlerfreiem Betrieb nach Westeuropa und wenn dies nicht der Fall ist, nach USA Osten.

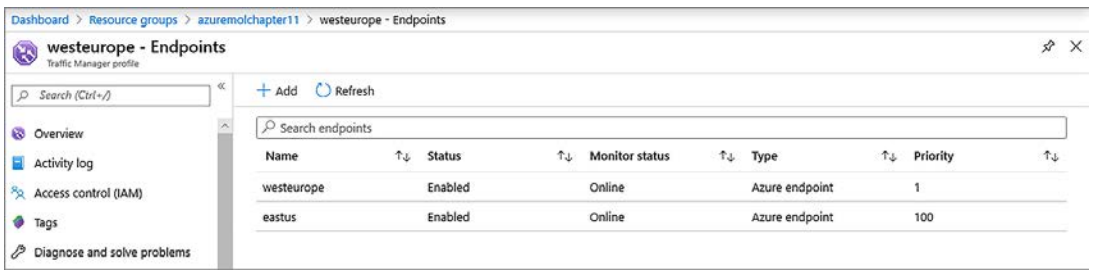


Abbildung 11.9 Die Endpunkte sind wie im vorherigen Traffic Manager-Profil konfiguriert. Diesmal mit umgekehrtem Standort der Web Apps. Diese untergeordneten Profile können Kunden immer auf die Web App in USA Osten oder Westeuropa leiten. Doch jetzt besteht eine Redundanz auf einen anderen Endpunkt, falls der primäre Endpunkt in der Region ausfällt.

Kommen wir nun zu dem letzten Teil in diesem Prozess. Wir betrachten eine bewährte Methode für hohe Verfügbarkeit bei Verwendung des Traffic Managers für die globale Verteilung von Anwendungen. In der Praxis ist Ihre Umgebung möglicherweise weniger komplex. Das Diagramm in Abbildung 11.10 verdeutlicht, welche untergeordneten Profile und Zuweisungen zu den regionalen Web Apps Sie erstellen müssen.

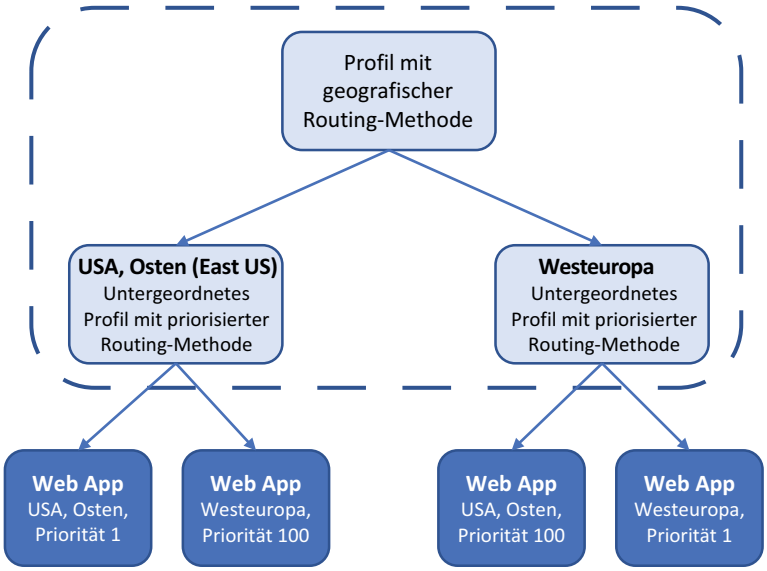


Abbildung 11.10 Sie haben die untergeordneten Traffic Manager-Profile für USA Osten und Westeuropa erstellt und die regionalen Web Apps und Prioritäten nach Bedarf konfiguriert. Nun müssen Sie die entsprechenden untergeordneten Profile mit dem übergeordneten verknüpfen.

Bei dem geografischen Routing definieren Sie ein Region, wie zum Beispiel USA Nord, und ein verschachteltes Profil wie eastus. Alle Kund*innen in Nordamerika werden auf dieses untergeordnete Profil weitergeleitet. Sie haben Prioritäten für dieses untergeordnete Profil festgelegt, daher bearbeitete USA Osten immer den Datenverkehr. Doch Sie haben eine Redundanz festgelegt, damit bei Bedarf die Web App in Westeuropa genutzt wird.

Das Umgekehrte gilt für Kunden in Westeuropa. Sie können einen weiteren Endpunkt für das übergeordnete Traffic Manager-Profil erstellen. Dieses Mal sollte die Region Europa mit dem Endpunkt und dem verschachtelten Profil `westeurope` verbunden werden. Der gesamte europäische Datenverkehr wird auf dieses Profil geleitet. Die Web App in Westeuropa dient immer als Webanwendung. Bei Problemen wird der Datenverkehr nach USA Ost umgeleitet.

Sollten Richtlinien oder Verfügungen zur Datensouveränität verbieten, dass dieser Datenverkehr in eine andere Region umgelenkt wird, können Sie die Einstellung der Endpunkte im Traffic Manager entsprechend anpassen. So könnten Sie, dem Beispiel aus einem früheren Kapitel folgend, mehrere Web App in Westeuropa erstellen. Sie gewährleisten dadurch, dass den Kund*innen mehrere Web App-Instanzen zur Verfügung stehen. Sollte Ihre Anwendung auf VMs laufen, können Sie hinter dem Load Balancer eine Skala festlegen, um eine ähnliche Redundanz zu definieren.

Jetzt testen

Dabei kommt es auf Ihren eigenen regionalen Standort an! Sollten Sie sich außerhalb einer in den Traffic Manager-Profilen aufgeführten geografischen Zuordnungen befinden, müssen Sie Ihre eigene Region auswählen, sonst können Sie nicht auf die Web App in der Übung am Ende des Kapitels zugreifen.

Gehen Sie folgendermaßen vor, um untergeordnete und übergeordnete Profile zu verbinden:

1. Gehen Sie im Azure Portal auf Ressourcengruppe.
2. Wählen Sie das übergeordnete Traffic Manager-Profil aus. In den früheren Beispielen haben wir es ed azuremol genannt.
3. Gehen Sie in der Navigationsleiste links im Profil auf Endpunkte und anschließend auf „Add“ (Hinzufügen).
4. Erstellen Sie einen Endpunkt, der das erste untergeordnete Profil verwendet. Legen Sie den Typ als Nested Endpoint fest, und geben Sie einen Namen wie `eastus` ein: Wählen Sie als Zielressource das Traffic Manager-Profil aus, das Sie für USA Osten erstellt haben.
5. Wählen Sie unter „Regional Grouping“ (Geografische Zuordnung) aus dem Dropdown-Menü Nordamerika/Zentralamerika/Karibik aus, und gehen Sie auf OK.
6. Wiederholen Sie diese Schritte für einen weiteren Endpunkt. Nennen Sie diesen Endpunkt „westeurope“, setzen Sie die Zielressource auf das untergeordnete Traffic Manager-Profil für Westeuropa und wählen Sie für die geografische Zuordnung aus dem Dropdown-Menü Europa aus.

   Ihre Endpunkte für das übergeordneten Profil listen nun die beiden untergeordneten Profile auf. Bei jedem Profil ist ein Endpunkt mit der entsprechenden geografischen Region verbunden (siehe Abbildung 11.11).

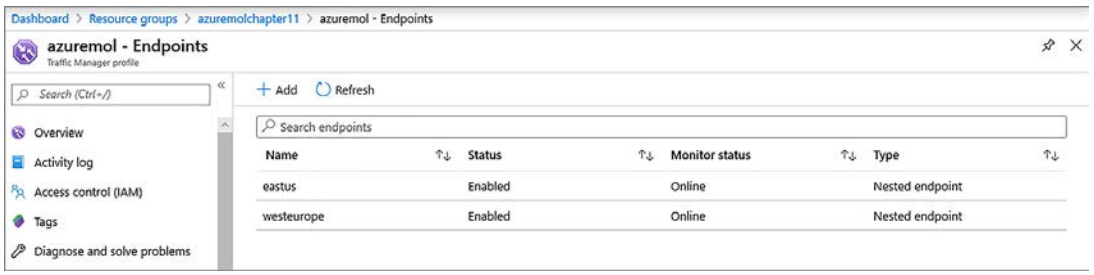


Abbildung 11.11 Verschachtelte untergeordnete Profile mit zugehörigen geografischen Regionen. Dieses übergeordnete Traffic Manager-Profil leitet nun den gesamten Datenverkehr von Europa auf die Web App in Westeuropa. Bei Problemen wird USA Osten angesteuert. Umgekehrt funktioniert es bei Kund*innen in Nordamerika/Zentralamerika/Karibik.

Die Web Apps sind derzeit so festgelegt, dass sie den Datenverkehr nur auf Ihrer Standarddomäne akzeptieren, das die Form *webappname*.azurewebsites.net aufweist. Leitet der Traffic Manager Kund*innen auf diese Web App-Instanzen, dann scheint der Datenverkehr aus der Domäne des übergeordneten Profils zu kommen, beispielsweise azuremol.trafficmanager.net. Da die Web Apps diese Domäne nicht erkennen, startet die Webanwendung nicht.

7 Fügen Sie die Domäne des übergeordneten Traffic Manager-Profils zu beiden, in den Schritten 4 bis 6 erstellten Web App-Instanzen hinzu. Den Domänennamen finden Sie auf der Übersichtsseite des übergeordneten Traffic Manager-Profils:

```
az webapp config hostname add \
  --resource-group azuremolchapter11 \
  --webapp-name azuremoleastus \
  --hostname azuremol.trafficmanager.net
az webapp config hostname add \
  --resource-group azuremolchapter11 \
  --webapp-name azuremolwesteurope \
  --hostname azuremol.trafficmanager.net
```

Öffnen Sie die Adresse Ihres übergeordneten Traffic-Manager-Profils in einem Webbrowser, zum Beispiel: https://azuremol.trafficmanager.net, you can't tell which endpoint you access, as both web apps run the same default web page. In der Übung am Ende dieses Kapitels laden Sie eine einfache Webseite auf jede Web App hoch, damit Sie diese unterscheiden können.

Fassen wir zusammen, was Sie in diesen Übungen getan haben. Das ist wichtig, da Ihre Kund*innen nun alle hochverfügbaren und redundanten Funktionen aus den früheren Kapiteln nutzen können, einschließlich des automatischen Traffic-Routings, das sie an die nächstgelegene Instanz Ihrer Webanwendung weiterleitet. In diesem Kapitel haben Sie Folgendes erstellt:

- Eine Web App in der Region USA Osten und eine weitere in Westeuropa
- Traffic Manager-Profile, die geografisches Routing nutzen, damit alle Kund*innen in Nord- und Zentralamerika auf die Web App in USA Osten und alle Kund*innen in Europa auf die Web App in Westeuropa geleitet werden.
- Untergeordnete Traffic Manager-Richtlinien mit Routing nach Priorität, damit eine alternative Region angesteuert werden kann, wenn die primäre Web App für die Region ausfällt.

Das bedeutet hinsichtlich der hohen Verfügbarkeit:

- Wenn Sie dieses Setup mit Web Apps mit automatischer Skalierung nutzen, entsteht sofort eine enorm hohe Redundanz.
- Kombinieren Sie diese Web Apps mit Cosmos DB, werden alle Ihre Anwendungen automatisch skaliert und global verteilt. Ihre Kund*innen greifen immer auf die ihnen nächstgelegenen Ressourcen zu, und Sie erzielen kürzeste Reaktionszeiten und beste Leistung.
- Und selbst wenn Sie an VMs festhalten, können Sie eine vergleichbare, hochverfügbare und global verteilte Umgebung bieten, in dem Sie Skalierungsgruppen mit Lastenausgleichsmodulen verwenden.

Und ja, Sie können Traffic Manager durch Front Door ersetzen, wenn Sie erweiterte Datenverkehrsverwaltungsfunktionen auf Anwendungsebene verwenden müssen.

In den letzten Kapiteln sind viele neue Dinge zur Sprache gekommen. Mit jedem Kapitel dürften Sie sich so ziemlich die ganze Mittagspause lang beschäftigt haben! Doch dafür hat sich in der vergangenen Woche auch ganz schön etwas getan. Sie können jetzt eine Webanwendung mit IaaS-VMs oder PaaS-Web-Apps erstellen, sie hochverfügbar machen, einen Lastenausgleich durchführen und automatische Skalierung anwenden (siehe Abbildung 11.12). Sie sind in der Lage Ihre Datenbankanforderungen mit einem global verteilten Cosmos DB-Backend zu erfüllen, und Sie können Kund*innen automatisch in die nächstgelegene regionale Instanz Ihrer Anwendung leiten, und das alles in mit Azure gehosteter DNS.

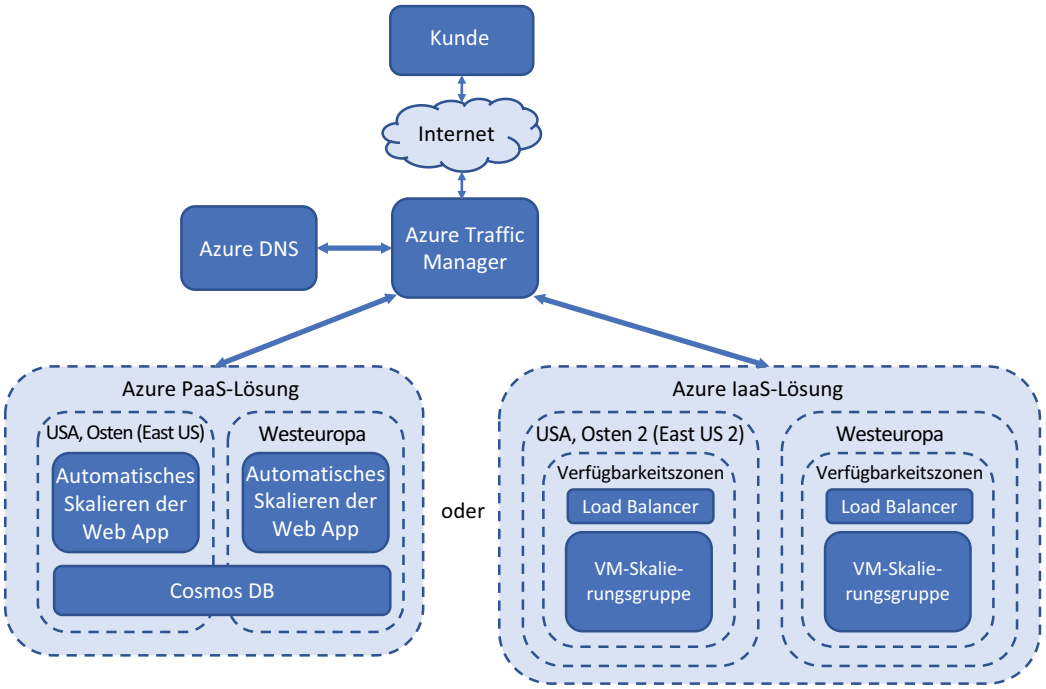


Abbildung 11.12 Nach den letzten Kapiteln sollte Ihnen klar sein, wie Sie hochverfügbare IaaS- oder PaaS-Anwendungen in Azure erstellen können. Die IaaS-Lösungen können Verfügbarkeitszonen, Load Balancer und Skalierungsgruppen nutzen. Die PaaS-Lösungen können Web Apps mit automatischer Skalierung und Cosmos DB verwenden. Traffic Manager und Azure DNS können Kunden anhand ihrer geografischen Region automatisch auf die optimale Anwendungsinstanz leiten.

In der Übung am Ende des Kapitels laden Sie einige einfache Webseiten auf Ihre Web Apps hoch, um zu überprüfen, ob Traffic Manager funktioniert und der richtige Endpunkt Ihren Datenverkehr bearbeitet. Machen Sie diese Übung, wenn Sie Zeit dazu haben. Oder gönnen Sie sich ansonsten eine wohlverdiente Pause. Das bleibt unter uns!

Im zweiten Abschnitt dieses Buchs gibt es noch ein weiteres Kapitel. Darin erfahren Sie, wie Sie Ihre Anwendungen betriebsbereit halten können: durch Überwachung und Fehlerbehebungen bei Ihren Anwendungen und in Ihrer Infrastruktur.

11.4 Übung: Bereitstellung von Webanwendungen, um Traffic Manager in Aktion zu sehen

Auch dieses Kapitel war sehr umfangreich. Sehen Sie diese Übung als ein mentales Fitnesstraining für Ihr Wissen über die Nutzung von Azure mit Web Apps. In den Beispielen zu Azure enthält die GitHub-Repository zwei einfache Webseiten für unsere Online-Pizzeria. In dem Titel jeder Webseite ist der Standort der Web App aufgeführt. Laden Sie diese Webseiten auf die entsprechende Web App-Instanz hoch, um die Datenströme Ihres Traffic Managers in der Praxis zu sehen:

1. Klonen Sie gegebenenfalls die GitHub-Repository-Beispiele wie folgt in Ihrer Cloud Shell:

   ```
   git clone https://github.com/fouldsy/azure-mol-samples-2nd-ed.git
   ```

2. Beginnen Sie mit der Website in USA Osten, und wiederholen Sie dann die folgenden Schritte für das Verzeichnis mit Westeuropa:

   ```
   cd ~/azure-mol-samples-2nd-ed/11/eastus
   ```

3. Initialisieren Sie das Git Repository, und fügen Sie die einfache Webseite hinzu:

   ```
   git init && git add . && git commit -m "Pizza"
   ```

4. Das Übersichtsfenster für Ihre Web App im Azure Portal zeigt die Git-Klon-URL an. Kopieren Sie diese URL, und stellen Sie sie mit folgendem Befehl als Ziel für Ihre HTML-Beispielseite in der Cloud Shell ein:

   ```
   git remote add eastus <your-git-clone-url>
   ```

5. Leiten Sie die HTML-Beispielsite per Push zu Ihrer Web App:

   ```
   git push eastus master
   ```

6. Wiederholen Sie diese Schritte für das Verzeichnis azure-mol-samples-2nd-ed/11/westeurope.
7. Wenn Sie fertig sind, öffnen Sie Ihren Webbrowser mit dem Domänennamen Ihres übergeordneten Traffic Manager-Profils, z. B. https://azuremol.trafficmanager.net, to see the traffic flow.

12 Überwachung und Fehlerbehebung

In den vorherigen Kapiteln haben Sie erfahren, wie Sie Ihre Anwendungen hochverfügbar machen und Kund*innen aus der ganzen Welt auf global verteilte Instanzen Ihrer Anwendung weiterleiten können. Ziel war es unter anderem, die Interaktion mit Ihrer Anwendungsinfrastruktur zu minimieren und die Integrität und Leistung Ihrer Azure-Plattform automatisch für Sie zu steuern. Doch manchmal ist es unvermeidlich, die Diagnose- oder Leistungsdaten zu überprüfen. In diesem Kapitel erfahren Sie, wie Sie die Startdiagnose für eine VM überprüfen, die Leistungsdaten überwachen und Verbindungsprobleme mit dem Network Watcher lösen können.

12.1 VM-Startdiagnose

Sie brauchen nur Ihren Code in Web Apps einzubinden, und die Azure-Plattform erledigt dann den Rest für Sie. In Kapitel 3 haben wir uns mit der Behandlung und Diagnose von Problemen bei Web App-Bereitstellungen beschäftigt. Sie haben erfahren, wie Sie Anwendungsereignisse in Echtzeit darstellen und die Leistung überwachen können. Wenn Sie mit VMs in der Cloud arbeiten, ist es oft schwierig, ein Problem zu beheben, wenn Sie den Computerbildschirm nicht physisch sehen können, wie es bei der Diagnose von Webanwendungen der Fall ist.

Fehlende Verbindungen gehören zu den häufigsten Problemen bei VMs. Wie können Sie Probleme behandeln, wenn Sie keine SSH- oder RDP-Verbindung zu einer VM haben? Als erstes sollten Sie überprüfen, ob die VM korrekt läuft. Dazu bietet Azure die VM-Startdiagnose an. Sie beinhaltet Bootlogs und einen Screenshot der Konsole.

Interaktiver Zugriff auf die Boot-Konsole

Für spezifische Szenarien zur Problembehandlung können Sie auch live auf eine serielle Konsole für VMs auf Azure zugreifen. Diese serielle Konsole ermöglicht interaktive Anmeldungen und die Problembehandlung bei Schwierigkeiten mit dem Booten. Sie können Ihre VM neu konfigurieren, um fehlerhafte Startszenarien oder Fehlkonfigurationen von Diensten und Anwendungen zu korrigieren, die ein fehlerfreies Booten Ihrer VM verhindern.

In diesem Kapitel finden Sie keine spezifischen Szenarien für die Verwendung der seriellen Konsole. Sie erhalten dabei nützliche Informationen, während Sie beim Bootvorgang praktisch vor dem Monitor einer VM sitzen. Die Startdiagnose muss aktiviert sein, daher sind diese Übungen Voraussetzung für die serielle Konsole.

Jetzt testen

Gehen Sie folgendermaßen vor, um eine VM zu erstellen und die Startdiagnose zu aktivieren:

1. Klicken Sie oben links im Azure Portal auf „Create a Resource“ (Ressource erstellen).
2. Suchen Sie ein Windows Server 2019 Datacenter VM-Image, und wählen Sie es aus.
3. Erstellen Sie eine Ressourcengruppe, z. B. `azuremolchapter12`, und wählen Sie dann die am besten geeignete nächstgelegene Azure-Region aus.
4. Wählen Sie eine VM-Größe, wie DS1_v2.
5. Geben Sie einen Benutzungsnamen für die VM ein, z. B. `azuremol`, und ein Kennwort. Das Kennwort muss mindestens 12 Zeichen lang sein und drei der folgenden Zeichen enthalten: einen Kleinbuchstaben, einen Großbuchstaben, eine Zahl und ein Sonderzeichen.
6. Akzeptieren Sie alle Optionen für Redundanz oder Regeln für Ports für eingehenden Datenverkehr.
7. Akzeptieren Sie die Standardwerte für Datenträger und Netzwerke. Sie müssen nichts ändern. Diese Einstellungen sollten Ihnen bereits vertraut sein.

   Den Abschnitt „Management“ haben Sie vielleicht früher übersprungen. Abbildung 12.1 verdeutlicht, dass die Option zur Startdiagnose standardmäßig aktiviert ist und ein Speicherkonto erstellt wird.
8. Lassen Sie die Option zur Diagnose des Gastbetriebssystems vorerst deaktiviert.
9. Überprüfen Sie Ihre VM-Konfigurationseinstellungen, und wählen Sie dann „Create“ (Erstellen).

Das Erstellen und Konfigurieren der VM dauert einige Minuten. Wenden wir uns also wieder der Startdiagnose zu.

Falls bei nicht aktivierter Startdiagnose ein Problem auftritt, können Sie die VM wahrscheinlich nicht booten, um die Diagnose zu aktivieren. Das wäre eine echte Zwickmühle. Deshalb ist die Startdiagnose für die im Azure Portal erstellten VMs automatisch aktiviert.. Für Azure PowerShell, Azure CLI und die sprachspezifischen SDKs müssen Sie die

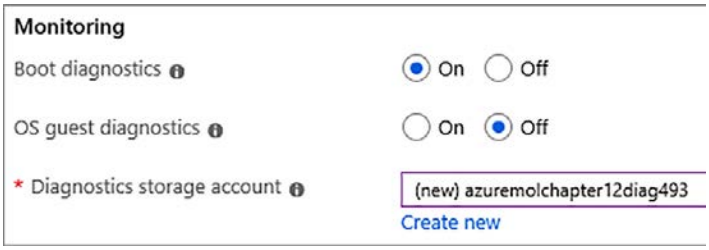


Abbildung 12.1 Die Startdiagnose ist standardmäßig aktiviert, wenn Sie eine VM im Azure Portal erstellen. Die Startdiagnosen werden in einem eigens eingerichteten Speicherkonto gespeichert. Die Diagnose von Gastbetriebssystemen überprüfen, und aktivieren Sie in einer späteren Übung, lassen Sie diese daher einstweilen deaktiviert. Für den Einsatz in der Produktion sollten Sie sowohl die Startdiagnose als auch die Diagnose des Gastbetriebssystem für jede von Ihnen erstellte VM aktivieren.

Startdiagnostik selbst aktivieren. Es empfiehlt sich auf jeden Fall, die Startdiagnosen beim Erstellen Ihrer VMs zu aktivieren. Verwenden Sie stets Azure Resource Manager-Vorlagen (Kapitel 6) oder Ihre eigenen Azure CLI- oder PowerShell-Skripts, die die Startdiagnose während der Bereitstellung ermöglichen.

Für die Bootprotokolle und Konsolen-Screenshots müssen Sie ein Speicherkonto erstellen. Die Speicherkosten für diese Daten betragen wahrscheinlich weniger als 0,01 USD pro Monat, es sei denn, Sie verfügen über eine viel genutzte VM, die viele Daten generiert. Wenn Sie erst einmal ein Problem mit einer VM haben, werden Sie merken, dass sich dieser Cent pro Monat für Sie rentiert. Dieses Speicherkonto kann auch zur Speicherung zusätzlicher Leistungskennzahlen und Protokolle auf VM-Ebene verwendet werden, die wir in Abschnitt 12.2 untersuchen werden. Auch dafür sind die Speicherkosten minimal. Selbst wenn Ihre VM-Umgebung wächst, sind die geringen zusätzlichen Kosten für eine schnelle Problembehandlung eine gute Investition.

Jetzt testen

Gehen Sie folgendermaßen vor, um die Startdiagnose für Ihre VM zu sehen:

1. Wählen Sie im Azure Portal links „Virtual Machines“ (Virtuelle Computer) aus.
2. Wählen Sie die in der vorherigen Übung erstellte VM aus.
3. Wählen Sie unter dem Abschnitt Support + Problembehandlung im VM-Menü „Boot diagnostics“ (Startdiagnose) aus. Die Startdiagnose und der VM-Status werden angezeigt (siehe Abbildung 12.2). Sollten Probleme beim Booten der VM auftreten, finden Sie einen entsprechenden Hinweis im Integritätsbericht, der für die Diagnose der Ursache des Problems hilfreich sein kann.

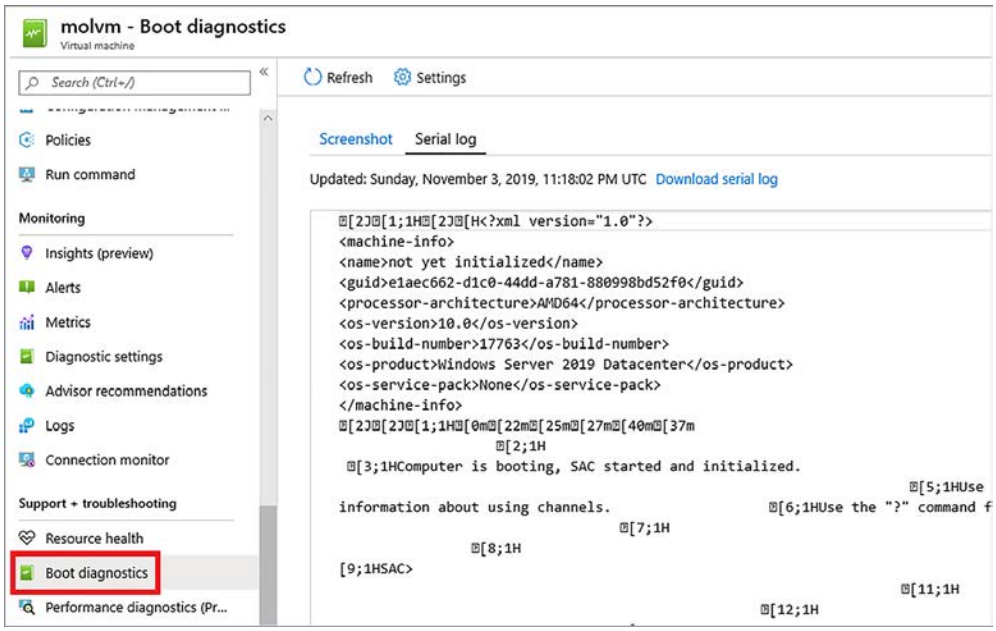


Abbildung 12.2 Die Startdiagnose für einen VM-Bericht enthält Informationen über die Integrität und den Boot-Status. Wenn Fehler angezeigt werden, sollte Sie deren Ursache erkennen und das Problem lösen können. Sie können die Protokolle zu Analysezwecken aus dem Portal auf Ihren lokalen Computer herunterladen.

12.2 Leistungsdaten und Warnmeldungen

Am Beginn einer Problembehandlung steht die Überprüfung der Leistung. Wie viel Speicher ist vorhanden, wie viel CPU wird verbraucht und hoch ist die Datenträgeraktivität?

Sie sollten, während Sie Ihre Anwendungen in Azure entwickeln und testen, an verschiedenen Stellen einen Ausgangswert für die Leistung aufzeichnen. Diese Ausgangswerte geben Ihnen eine Vorstellung davon, wie Ihre Anwendung bei unterschiedlicher Auslastung funktionieren soll. Warum ist das wichtig? Wie können Sie über einen Zeitraum von drei Monaten feststellen, ob Sie Leistungsprobleme haben, wenn keine Vergleichsdaten für die aktuelle Leistung vorliegen?

Beim Thema automatische Skalierung von Anwendungen in Kapitel 9 haben Sie einfache Leistungsdaten wie die CPU-Nutzung verwendet, um die Azure-Plattform so einzurichten, dass sie die Anzahl der Instanzen Ihrer Anwendung zu einem bestimmten Zeitpunkt erhöht oder verringert. Diese einfachen Messdaten geben Ihnen nur einen kleinen Einblick in die Leistung der VM. Aufschluss über detailliertere Kennzahlen gibt die Leistung der VM. Zu diesem Zweck benötigen Sie die Azure-Diagnoseerweiterung.

12.2.1 Anzeigen der Leistungsdaten mit der VM-Diagnoseerweiterung

Azure verfügt über Dutzende Erweiterungen, die Ihren VMs eine größere Funktionalität verleihen. Sie lassen sich problemlos installieren. Diese Erweiterungen installieren einen kleinen Agenten oder eine Anwendungslaufzeit auf der VM, die häufig Informationen an die Azure-Plattform oder Lösungen von Drittanbietern meldet. VM-Erweiterungen können automatisch Komponenten konfigurieren und installieren oder Skripte auf Ihren VMs ausführen.

Die VM-Diagnose-Erweiterung wird verwendet, um Leistungsdaten aus der VM auf ein Speicherkonto zu übertragen. Diese Leistungsdaten können anschließend im Azure Portal analysiert oder in einer bestehenden Überwachungslösung heruntergeladen und genutzt werden. Die Diagnoseerweiterung ermöglicht Ihnen ein tieferes Verständnis der CPU-Leistung und des Speicherverbrauchs aus der VM. Sie erhalten in der Regel ein detaillierteres und genaueres Bild, als der Host es liefern kann.

Automation und VM-Erweiterungen

In Kapitel 18 behandeln wir das Thema Azure Automation. Sie können damit automatisierte und planmäßige Aufgaben an Ihren VMs erledigen. Eine leistungsstarke Funktion von Azure Automation ist der Einsatz als PowerShell Desired State Configuration (DSC) Pullserver. PowerShell DSC dient dazu, einen bestimmten Status festzulegen, der besagt, wie ein System konfiguriert werden soll, welche Pakete zu installieren sind, welche Dateien und Berechtigungen usw. zu verwenden sind. Sie erstellen Definitionen für die gewünschte Konfiguration und wenden sie auf VMs oder physikalische Server an. Sie können dann über die Einhaltung dieser Richtlinien berichten und diese durchsetzen. Die Azure PowerShell DSC-Erweiterung wird verwendet, um DSC-Konfigurationen, die zum Beispiel von einem Azure Automation Pullserver stammen, einzusetzen.

Es gibt noch mehr Erweiterungen, die Konfigurationen anwenden und Skripte auf VMs ausführen können. Ein Beispiel ist die Azure Custom Script Extension. Mit der Custom Script Extension definieren Sie entweder einen einfachen Satz von Befehlen oder verweisen auf ein oder mehrere externe Skripte, wie sie etwa in Azure Storage oder GitHub gehostet werden. Diese Skripte können komplexe Konfigurations- und Installationsaufgaben ausführen. Sie gewährleisten, dass alle eingesetzten VMs einheitlich konfiguriert sind.

Sowohl die Azure PowerShell DSC-Erweiterung als auch die Custom Script Extension werden häufig mit VM-Skalierungsgruppen verwendet. Nachdem Sie eine dieser Erweiterungen für die Skalierungsgruppe eingerichtet haben, erstellen Sie in der Skalierungsgruppe VM-Instanzen. Diese werden automatisch so konfiguriert, dass sie Ihre Anwendung ausführen. Diese Erweiterungen sollen die erforderliche manuelle Konfigurierung von VMs minimieren, denn das ist ein fehleranfälliger Prozess, der menschliches Handeln erfordert.

Weitere Möglichkeiten, VM-Konfigurationen zu automatisieren, sind Puppet und Chef. Zu beiden gibt es Azure VM-Erweiterungen. Wenden Sie sich, wenn Sie bereits ein Konfigurationsmanagement-Tool verwenden, an Ihren Anbieter, um zu erfahren welche Art der Nutzung von Azure er unterstützt. Es ist gut möglich, dass es eine VM-Erweiterung gibt, die Ihnen das Leben leichter macht.

Jetzt testen

Gehen Sie folgendermaßen vor, um die VM-Diagnose-Erweiterung zu aktivieren:

1. Wählen Sie im Azure Portal links „Virtual Machines“ (Virtuelle Computer) aus.
2. Wählen Sie die in einer früheren Übung erstellte VM aus.
3. Wählen Sie unter dem Abschnitt „Überwachung“ im VM-Menü die Diagnose-Einstellungen aus.
4. Wählen Sie den Button Überwachung auf Gastebene aktivieren.

Bis die Überwachung auf Gastebene aktiviert ist, können einige Minuten vergehen. Hinter den Kulissen tut Azure genau das:

- Installation der VM-Diagnose-Erweiterung
- Konfiguration der Erweiterung, um Metriken auf Gastebene für folgende Bereiche zu streamen:
  - Logischer Datenträger
  - Arbeitsspeicher
  - Netzwerkschnittstelle
  - Prozess
  - Prozessor
  - System
  - Ermöglicht das Streaming von Anwendungs-, Sicherheits- und Systemprotokollen auf Azure Storage

Sobald die Diagnose-Erweiterung installiert ist, können Sie festlegen, welche Daten gesammelt werden, indem Sie nur bestimmte Leistungsindikatoren für den Bericht auswählen. Sie können zum Beispiel nur die Speichernutzung sammeln oder das Sammeln von Microsoft SQL Server-Metriken aktivieren. Metriken werden standardmäßig alle 60 Sekunden erfasst. Sie können diese Abtastrate nach Wunsch für Ihre Anwendungen und Infrastruktur anpassen.

Die VM-Diagnose-Erweiterung kann auch Logdateien aus Ihrer VM streamen., sodass Sie die Anwendungs-, Sicherheits- und Systemprotokolle für Analysen oder Warnungen zentralisieren können, wie in Abbildung 12.3 dargestellt. Anwendungs- und Systemprotokolle, die kritische Meldungen, Fehler oder Warnungen ausgeben, werden standardmäßig zusammen mit Sicherheitsereignissen für Audit Failure protokolliert. Sie können die aufzuzeichnenden Protokollebenen sowie die Sammlung von IIS-Protokollen, Anwendungsprotokollen und die Ereignisse im Event Tracing for Windows (ETW) ändern. Im Rahmen der Planung und Bereitstellung Ihrer Anwendung bestimmen Sie, welche Protokolle Sie sammeln möchten.

Windows-VMs sind in dieser Hinsicht nichts Besonderes. Sie können die Diagnoseerweiterung für Linux-VMs auf die gleiche Weise verwenden, um Leistungsdaten zu erhalten und verschiedene Protokolle zu streamen.

Die häufig einzige Möglichkeit zur Analyse der Ereignisse bei Problemen Ihrer VM besteht darin, die *Absturzabbilder* zu überprüfen. Die Support-Kanäle fordern diese Dumps häufig an, wenn man der Ursache eines Problems auf den Grund gehen will. Wie bei der Boot-Diagnose gibt es keine Möglichkeit, nachträglich Crash-Dumps zu aktivieren, um herauszufinden, warum etwas fehlgeschlagen ist. Legen Sie also fest, ob Sie bestimmte Prozesse überwachen müssen, und konfigurieren Sie proaktiv Crash-Dumps. Sie könnten beispielsweise den IIS-Prozess überwachen und einen vollständigen Crash-Dump in Azure Storage aufzeichnen, wenn der Prozess fehlschlägt.

Es gibt einige weitere Bereiche, die Sie für Gastmetriken konfigurieren können:

- *Sinks* ermöglicht es Ihnen, die VM-Diagnose-Erweiterung zum Versenden bestimmter Ereignisse an Azure Application Insights zu konfigurieren. Mit Application Insights erhalten Sie direkte Einsicht in die Leistung Ihres Codes.
- *Agent* ermöglicht es Ihnen, ein Speicherkontingent für alle Ihre Metriken festzulegen (die Voreinstellung beträgt 5 GB). Sie können auch die Sammlung von Protokollen für den Agenten selbst aktivieren oder den Agenten deinstallieren.

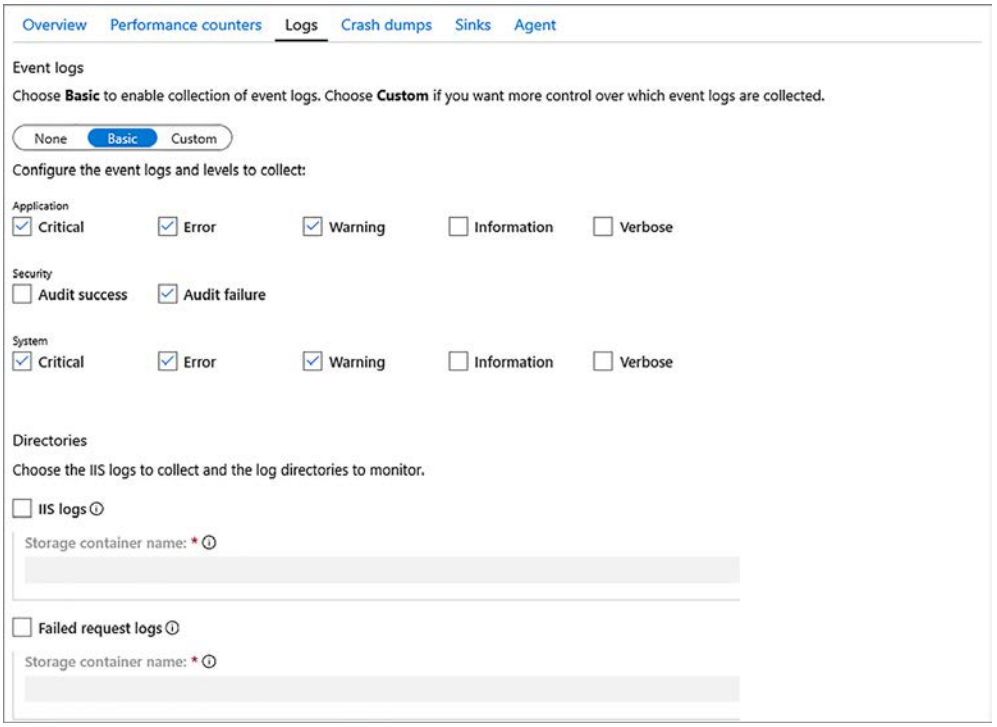


Abbildung 12.3 Sie können Ereignisse und Protokollebenen für verschiedene Komponenten innerhalb der VM konfigurieren. Damit können Sie Ihre VM-Protokolle zur Analyse und zur Erstellung von Benachrichtigungen zentralisieren. Kommt es zu Problemen mit Ihren Azure VMs, können Sie ohne die Installation komplexer und häufig kostspieliger Überwachungssysteme Benachrichtigungen überprüfen und erhalten.

Jetzt testen

Gehen Sie folgendermaßen vor, um die Metriken auf Gastebene zu sehen:

1. Wählen Sie im Azure Portal links „Virtual Machines“ (Virtuelle Computer) aus.
2. Wählen Sie die in einer früheren Übung erstellte VM aus.
3. Wählen Sie im Abschnitt „Monitoring“ des VM-Menüs „Metrics“ (Metriken) aus.

Im Vergleich zu den grundlegenden hostbasierten Metriken aus Kapitel 9 sind jetzt viele weitere Metriken verfügbar. Entdecken Sie einige der verfügbaren VM-Host- und Gast-Metriken, und denken Sie an einige Anwendungen, für die Sie bestimmte Metriken überwachen möchten.

12.2.2 Erstellen von Benachrichtigungen für Leistungsbedingungen

Wie erfahren Sie von einem Problem, wenn Ihre VM so konfiguriert ist, dass Leistungsmetriken auf Gastebene angezeigt werden? Sie werden wohl kaum dasitzen und die Leistungsgrafiken beobachten wollen, bis ein Problem auftaucht! Wenn doch, dann ist das Ihre Sache. Aber es gibt eine viel bessere Methode: Metrikwarnungen.

Metrikwarnungen ermöglichen Ihnen die Auswahl einer Ressource, einer Metrik und eines Schwellenwertes. Definieren Sie anschließend, wer wie benachrichtigt werden soll, wenn ein Schwellenwert erreicht ist. Warnmeldungen funktionieren nicht nur bei VMs. Sie können Warnmeldungen für öffentliche IP-Adressen definieren, die nach eingehenden Distributed-Denial-of-Service-Paketen (DDoS) Ausschau halten und Sie warnen, wenn ein bestimmter Schwellenwert erreicht ist, der einen Angriff darstellen könnte.

Sie können Warnmeldungen per E-Mail an Eigentümer*innen, Mitwirkende und Leser senden lassen. Grundlage für die Erfassung dieser Benutzer*innen und E-Mail-Adressen sind die RBAC-Richtlinien. In größeren Organisationen könnte dies den E-Mail-Versand an sehr viele Personen bedeuten. Nutzen Sie diese Funktion daher mit Vorsicht! Oder Sie geben bestimmte E-Mail-Adressen an, etwa die der Anwendungsinhaber oder bestimmter Infrastrukturingenieur*innen, oder auch eine Verteilerliste oder Gruppe, die sich an die direkt Beteiligten richtet.

Es gibt weitere nützliche Reaktionsmöglichkeiten, wenn ein Alarm ausgelöst wird:

- *Führen Sie ein Runbook aus.* In Kapitel 18 untersuchen wir Azure Automation. Mit dem Automatisierungsdienst können Sie Runbooks, die Skripte ausführen, erstellen und verwenden. Diese Skripte können einfache Abhilfemaßnahmen für die VM ausführen, wie zum Beispiel einen Prozess oder sogar die VM neu starten. Sie können auch Azure PowerShell-Cmdlets verwenden, um Azure Network Watcher-Funktionen, wie beispielsweise das Erfassen von Paketen, zu aktivieren, die wir am Ende dieses Kapitels untersuchen.
- *Führen Sie eine Logic App aus.* Mit Azure Logic Apps können Sie Workflows erstellen, die Serverless Code ausführen. Sie können Informationen in ein Support-Ticket-System eingeben oder einen automatisierten Telefonanruf an einen Techniker im Bereitschaftsdienst starten. Im Kapitel 21 erforschen wir die wunderbare Welt des Serverless Computing mit Azure Logic Apps und Azure Functions.

In der Übung am Ende des Kapitels konfigurieren Sie einige Warnmeldungen für Ihre VM. Doch Azure kann weit mehr, als nur die Problembehandlung und Überwachung Ihrer VMs. Wenden wir uns jetzt einer weiteren Fehlerquelle zu: dem Netzwerk.

12.3 Azure Network Watcher

VM-Leistungsdaten und Startdiagnosen eignen sich hervorragend zur Überwachung Ihrer Azure IaaS-Anwendungen. Web App-Anwendungsprotokolle und App Insights verdeutlichen die Leistungsfähigkeit Ihrer PaaS-Anwendungen. Hinter Konnektivitätsproblemen, die Sie oder Ihre Kund*innen bei einer Anwendung erleben, steckt nicht allzu selten der Netzwerkverkehr.

In Kapitel 5 habe ich noch darüber gewitzelt, dass die Netzwerkverantwortlichen immer für Probleme verantwortlich gemacht werden, die sich das Betriebsteam nicht erklären kann. An dieser Stelle können sie Frieden miteinander schließen, oder zumindest können wir den Beweis liefern, dass das Netzwerk schuld ist! Azure Network Watcher ist in dieser Hinsicht ein wahrer Friedensstifter. Mit Network Watcher können Sie folgende Funktionen überwachen und Probleme beheben:

- Erfassen von Netzwerkpaketen
- Validierung des IP-Datenflusses für NSGs
- Aufbau einer Netztopologie

Diese Funktionen haben die tolle Eigenschaft, mehrere Teams an der Lösung von Problemen zu beteiligen. Wenn Sie nach dem Erstellen einiger MVs keine Verbindung zu ihnen erhalten, können Sie die Netzwerkverbindung prüfen. Wenn Sie als

Entwickler*in Ihre Anwendung nicht mit dem Datenbank-Backend verbinden können, haben Sie die Möglichkeit, die NSG-Regeln auf ein Problem hin zu überprüfen. Netzwerkingenieure können Pakete erfassen, um den kompletten Kommunikationsfluss zwischen den Rechnern ausführlicher zu analysieren.

Weitere Problembehandlung bei Netzwerken

Network Watcher nutzt die in diesem Kapitel oben erwähnten Diagnoseprotokolle und Metriken. Auch Netzwerkressourcen wie Load Balancer und Anwendungsgateways können Diagnoseprotokolle erstellen. Diese Protokolle funktionieren genauso wie Anwendungs- und Systemprotokolle aus einer VM oder einer Web App. Die Protokolle werden im Azure Portal zusammengeführt. Dort können Sie herausfinden, ob es in der Konfiguration oder Kommunikation zwischen Hosts und Anwendungen Fehler gibt.

Auch DNS und Traffic Manager haben im Azure Portal einen Bereich für die Problembehandlung. Das Portal führt Sie durch einige häufige Fehler, denen Sie begegnen können. Sie finden dort Informationen über die Konfiguration und Links zu zusätzlichen Dokumentationen. Wenn all das nichts hilft, können Sie beim Azure Support eine Supportanfrage stellen.

Auch wenn es oft einfacher sein kann, große Anwendungen mit Azure Resource Manager-Vorlagen oder mit Azure-CLI-oder PowerShell-Skripten zu erstellen, bietet das Azure Portal doch viele sehr nützliche Tools und Funktionen, wenn etwas schief geht. Vor allem bei komplizierten Netzwerkkonfigurationen und Sicherheitsrichtlinien reich oft eine kurze Überprüfung der Ergebnisse aus den Network Watcher-Tools, um ein Problem zu erkennen und schnell zu lösen. Mit diesen Tools verbessern Sie die allgemeine Integrität und optimieren das Benutzererlebnis Ihrer Kund*innen.

Bei welchen Szenarien wäre der Network Watcher und seine Funktionen zur Problembehandlung eine echte Hilfe für Sie? Betrachten wir nun einige übliche Probleme und die von Network Watcher angebotene Lösungsstrategie.

12.3.1 IP-Datenflussüberprüfung

Ein ganz alltägliches Problem: Kund*innen haben keine Verbindung zu Ihrer Anwendung. Wenn Sie sich von Ihrem Büro aus verbinden funktioniert alles tadellos, aber Kunden können nicht über das öffentliche Internet auf die Anwendung zugreifen. Warum?

VPNs und ExpressRoute

Azure VPN (Virtuelle Private Netzwerke) schafft eine sichere Kommunikation zwischen den Büros On-Premises und den Azure Rechenzentren. Azure ExpressRoute schafft schnelle, spezielle private Verbindungen von On-Premises-Standorten zu den Azure-Rechenzentren. Es wird häufig in großen Unternehmen eingesetzt.

Beide Verbindungen sind recht komplizierter einzurichten. Unsere Zeit in einer einzigen Mittagspause reicht dafür nicht aus. Zudem sind sie wahrscheinlich nur einmal einzurichten und zu konfigurieren. Das Netzwerk-Team ist in der Regel für diese Konfiguration verantwortlich. Sie werden es wahrscheinlich nicht einmal merken, dass Sie Azure über ein VPN nutzen.

Alle Tests Ihrer Anwendung funktionieren hervorragend. Sie können über einen Webbrowser auf die Anwendung zugreifen, Bestellungen aufgeben und E-Mail-Benachrichtigungen erhalten. Wenn aber Ihre Kund*innen eine Bestellung aufgeben wollen, wird die Anwendung nicht geladen.

Was kann Network Watcher hier tun? Den IP-Datenfluss überprüfen. Network Watcher simuliert den Datenverkehr zu Ihrem Ziel und berichtet, ob der Datenverkehr erfolgreich zu Ihrem VM gelangen kann.

Jetzt testen

Gehen Sie folgendermaßen vor, um Netzwerk-Watcher zu aktivieren und den IP-Datenfluss zu überprüfen:

1. Wählen Sie im Azure Portal im Navigationsmenü links „All Resources“ (Alle Ressourcen) aus.
2. Suchen Sie in der Liste der verfügbaren Dienste Network Watcher aus. Aktivieren Sie Network Watcher in den Regionen, die Sie überwachen möchten. Wenn Sie Network Watcher in einer Region aktivieren, verwendet Azure für die verschiedenen Ressourcen und den Netzwerkverkehr die rollenbasierte Zugriffssteuerung.
3. Erweitern Sie die Liste der Regionen für Ihr Konto. Einige Regionen sind möglicherweise bereits aktiviert. Wenn die Region, in der Ihre VM bereitgestellt wurde, nicht aktiviert ist, wählen Sie die Region aus, und aktivieren Sie dann den Network Watcher.
4. Wenn Network Watcher in einer Region aktiviert ist (es dauert ein oder zwei Minuten), wählen Sie unter „Network Diagnostic Tools“ (Netzwerkdiagnosetools) auf der linken Seite des Network Watcher-Fensters die Option „IP Flow Verify“ (IP-Flussüberprüfung).
5. Wählen Sie Ihre Ressourcengruppe, zum Beispiel azuremolchapter12, und die VM, zum Beispiel molvm aus. Das Protokoll ist standardmäßig auf TCP gesetzt und die Richtung ist eingehend. Die lokale IP-Adresse des virtuellen NIC ist ebenfalls ausgefüllt.
6. Geben Sie als lokalen Port den Wert 80 ein. Als sie beim Erstellen der VM in der vorherigen Übung die Standardeinstellungen akzeptierten, haben Sie Port 80 nicht geöffnet. Dies ist daher ein guter Test zum zu sehen, was passiert, wenn der Datenverkehr verweigert wird.
7. Geben Sie als Remote-IP-Adresse `8.8.8.8` ein. Diese Adresse mag ihnen bekannt vorkommen. Es handelt sich um einen von Google bereitgestellten offenen DNS-Server. Sie machen nichts mit diesem Server. Network Watcher braucht lediglich eine externe IP-Adresse, um den Datenverkehrsfluss zu simulieren. Sie können auch https://whatsmyip.com und Ihre echte öffentliche IP-Adresse eingeben.
8. Setzen Sie den Remote-Port auf den Wert 80, und wählen Sie dann „Check“ (Aktivieren) aus.

Das Ergebnis Ihrer IP-Datenflussprüfung sollte `Zugriff verweigert` lauten. Network Watcher informiert Sie anschließend darüber, welche Regel den Datenfluss stoppte: die Regel DefaultInboundDenyAll. Nun wissen Sie, dass eine Netzwerksicherheitsregel den Datenverkehr blockiert. Aber wo wird diese Regel angewendet? Im Subnetz, der virtuellen NIC oder der Anwendungssicherheitsgruppe? Network Watcher hat eine weitere Funktion, in der Sie dies erfahren!

12.3.2 Anzeigen von geltenden NSG-Regeln

Sie können NSG-Regeln auf einen einzigen virtuellen NIC, auf Subnetz-Ebene oder auf eine Gruppe von VMs in einer Anwendungssicherheitsgruppe anwenden. Es

werden Regeln kombiniert. Dadurch können Sie über ein ganzes Subnetz ein gemeinsames Regelwerk festlegen und dann für Anwendungssicherheitsgruppen (wie „TCP-Port 80 auf allen Webservern freigeben“) oder für eine einzelne VM weitere Einzelheiten benennen.

Die folgenden Beispiele erläutern, wie NSG-Regeln angewendet werden können:

- *Subnetzebene* – TCP-Port 5986 für die sichere Remote-Verwaltung aus dem Management-Subnetz 10.1.10.20/24 freigeben
- *Ebene der Anwendungssicherheitsgruppe* – TCP-Port 80 für HTTP-Datenverkehr zu Webanwendungen freigeben und Anwendungssicherheitsgruppe auf alle VMs der Webanwendungen anwenden
- *Ebene des virtuelle NIC* – TCP-Port 3389 für den Remote-Desktop-Zugriff aus dem Management-Subnetz 10.1.10.20/24 freigeben

Das sind Grundregeln, die explizit einen bestimmten Datenverkehr zulassen. Wenn keine *allow*-Regeln mit einem Netzwerkpaket übereinstimmen, tritt die standardmäßige *DenyAll*-Regel in Kraft, die den Datenverkehr stoppt.

Während der in diesem Beispiel behandelten Prüfung der Anwendung haben Sie möglicherweise die HTTP-Regel konfiguriert, die besagt, dass nur Datenverkehr aus einem Ihrer Subnetze am Firmenstandort zulässig ist. Deshalb haben Kund*innen über das öffentliche Internet keine Verbindung mehr.

Jetzt testen

Führen Sie die folgenden Schritte aus, um festzustellen, wo eine NSG-Regel angewendet wird:

1. Wählen Sie in Network Watcher auf der linken Seite „Effective security rules“ (Geltende Sicherheitsregeln) aus.
2. Wählen Sie Ihre Ressourcengruppe, wie zum Beispiel azuremolchapter12 und Ihre VM, wie zum Beispiel molvm. Es dauert einige Sekunden, bis die effektiven Regeln angezeigt werden (siehe Abbildung 12.4).

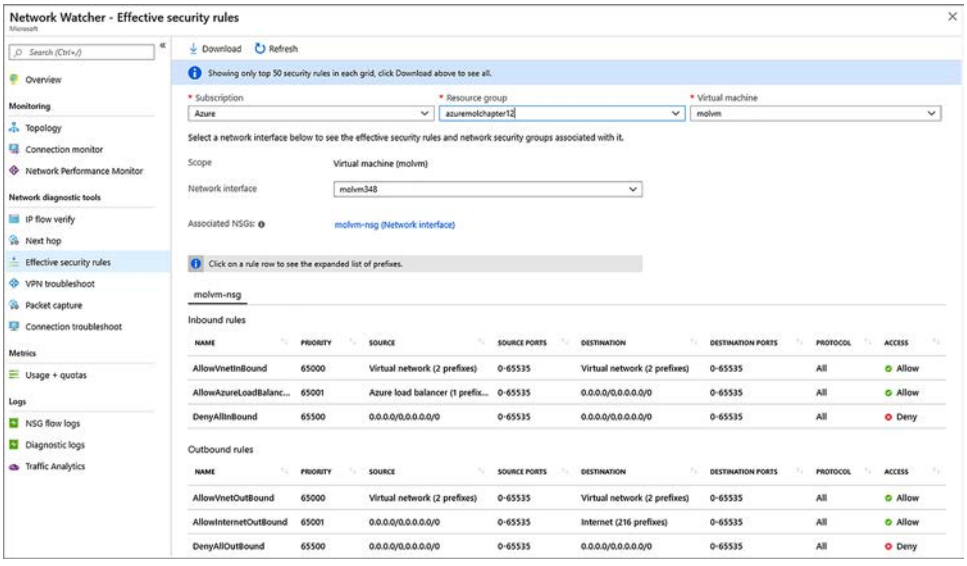


Abbildung 12.4 Wenn Sie eine VM auswählen, prüft Network Watcher, wie alle NSG-Regeln angewendet werden, in welcher Reihenfolge dies geschieht und welche effektive Regeln derzeit gelten. Sie können dann schnell zu den Subnetz-, virtuellen NIC- und Standardregeln gelangen, um zu erfahren, wo eine bestimmte Regel angewendet wird.

Die zuvor erstellten Standardregeln der VM sind nicht sehr aufregend. Sie können jedoch durch die Subnetz-, Netzwerkschnittstellen- und Standardregeln navigieren, um ein Gefühl für die Kombination effektiver Sicherheitsregeln zu bekommen. Sie erfahren, wie Sie herausfinden können, wo Regeln angewendet werden, wenn Sie Änderungen vornehmen müssen.

12.3.3 Erfassen von Netzwerkpaketen

Gehen wir einmal davon aus, dass Sie Ihre Netzwerksicherheitsregeln aktualisiert haben, damit Kund*innen aus dem öffentlichen Internet auf Ihre Anwendung zugreifen können. Aber ein Kunde berichtet, dass er ein merkwürdiges Verhalten feststellt. Die Webanwendung lädt manchmal nicht, oder zeigt Bilder nicht an. Die Verbindung kommt häufig aufgrund eines Timeouts nicht zustande.

Zeitweise auftretende Probleme zu lösen, ist häufig eine sehr schwierige Aufgabe. Das gilt vor allem, wenn Sie nur einen begrenzten oder gar keinen Zugriff auf den problembehafteten Computer haben. Ein gängiger Ansatz zur Problembehandlung ist die Erfassung der Netzwerkpakete und deren Überprüfung auf Probleme wie Netzwerkübertragungsfehler, fehlerhafte Pakete oder Protokolle und Kommunikationsprobleme.

Beim Erfassen von Netzwerkpaketen erhalten Sie den rohen Datenstrom zwischen zwei oder mehr Hosts. Die Kunst der Paketerfassung ist nichts für schwache Nerven! Spezielle Tools von Drittanbietern – genannt seien Wireshark von Riverbed, Fiddler von Telerik und Message Analyzer von Microsoft –

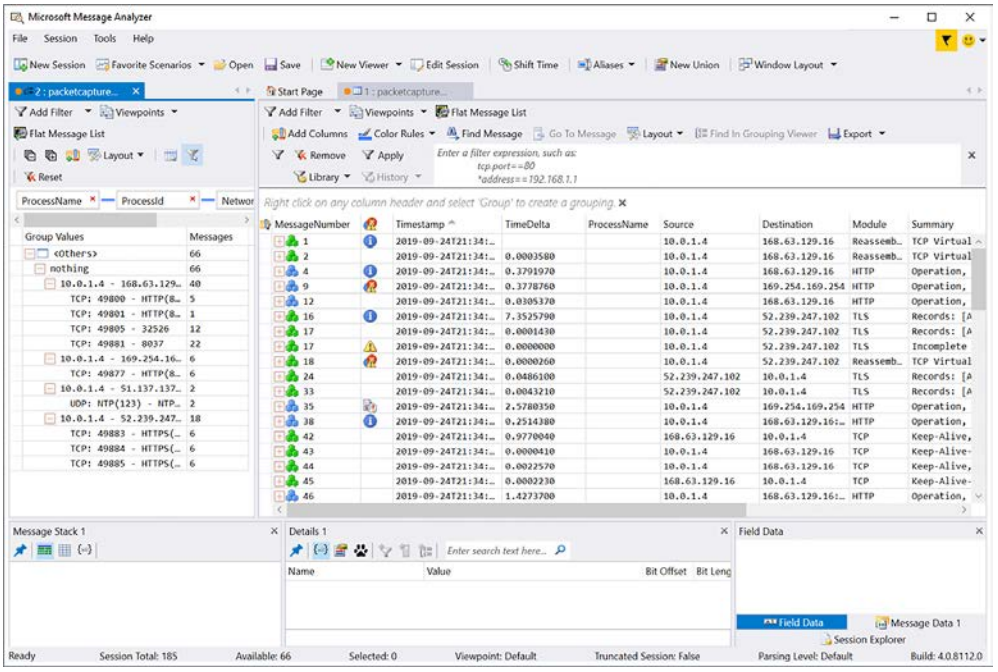


Abbildung 12.5 Netzwerkpaketerfassung im Message Analyzer von Microsoft. Jedes einzelne Paket steht zur Prüfung zur Verfügung. Sie können die Pakete nach Kommunikationsprotokoll oder Client-Host gruppieren und filtern. Bei dieser Tiefe können Sie die zwischen den Knoten tatsächlich fließenden Netzwerkdatenpakete analysieren und auftretende Fehler beheben. Ein ehemaliger Kollege sagte einmal zu mir: „Pakete lügen nie“. Das Rätsel besteht darin herauszufinden, was die Pakete verraten.

Message Analyzer bieten Ihnen eine grafische Möglichkeit, Netzwerkpakete zu sehen und zu filtern. Die Pakete werden dabei zumeist nach Kommunikationsmittel oder Protokoll gruppiert. Abbildung 12.5 zeigt ein Beispiel für die Erfassung eines Netzwerkpaketes.

Damit Netzwerk Watcher Pakete von und zu Ihren VMs erfassen kann, benötigen Sie zunächst die VM-Erweiterung für den Network Watcher. Sie haben im Abschnitt 12.3.2 gesehen, wie die Azure-Plattform mit VM-Erweiterungen in das Innere einer VM gelangen kann, um verschiedene Verwaltungsarbeiten durchzuführen. Die VM-Erweiterung für den Network Watcher untersucht den Netzverkehr von und zur VM.

Jetzt testen

Gehen Sie folgendermaßen vor, um die VM-Erweiterung für den Network Watcher zu installieren und Netzwerkpakete zu erfassen:

1. Klicken Sie im Azure Portal im Menü links auf „Virtual Machines“ (Virtuelle Computer), und wählen Sie dann Ihre VM aus, wie zum Beispiel molvm.
2. Wählen Sie in der Kategorie „Settings“ (Einstellungen) links im VM-Fenster die Option „Extensions“ (Erweiterungen).
3. Wählen Sie „Add an Extension“ (Erweiterung hinzufügen).
4. Wählen Sie in der Liste der verfügbaren Erweiterungen Network Watcher-Agent for Windows, und klicken Sie dann auf „Create“ (Erstellen).
5. Gehen Sie auf OK, um die Installation der Erweiterung zu bestätigen. Es kann einige Minuten dauern, bis der Network Watcher Agent auf Ihrer VM installiert ist.
6. Wählen Sie Navigationsmenü oben links im Portal die Option „All Resources“ (Alle Dienste) aus, um zum Network Watcher-Menü im Azure Portal zurückzukehren, und klicken Sie dann auf Network Watcher.
7. Wählen Sie in der Kategorie „Network Diagnostic Tools“ (Netzwerkdiagnosetools) links im Network Watcher-Fenster die Option „Packet Capture“ (Paketerfassung), und fügen Sie eine neue Erfassung hinzu.
8. Wählen Sie Ihre Ressourcengruppe, wie zum Beispiel azuremolchapter12 und die VM, wie zum Beispiel molvm. Geben Sie eine Bezeichnung für Ihre Paketerfassung ein, zum Beispiel `molcapture`.

   Paketerfassungen werden standardmäßig in Azure Storage gespeichert. Sie können auch In Datei speichern auswählen und einen lokalen Dateipfad in der Quell-VM angeben. Die VM-Erweiterung für den Network Watcher-Agenten schreibt dann die Datei für die Paketerfassung auf den Datenträger in der VM.
9. Wählen Sie, falls noch nicht aktiviert, den Namen des Speicherkontos aus, der mit dem Namen Ihrer Ressourcengruppe beginnt, wie zum Beispiel azuremolchapter12diag493. Es handelt sich dabei um das Speicherkonto, das von der VM-Diagnose-Erweiterung erstellt und verwendet wird, die Sie früher aktiviert haben.
10. Sie können eine maximale Größe für jedes Paket angeben (Standard ist 0 für das gesamte Paket), die maximale Dateigröße für die Paketerfassungssitzung (Standard ist 1 GB) und die zeitliche Begrenzung für die Paketerfassung (Standard ist 5 Stunden). Um nur den Datenverkehr aus bestimmten Quellen oder von bestimmten Ports zu erfassen, können Sie einen Filter hinzufügen, der den Umfang Ihrer Paketerfassungen eingrenzt.

11 Legen Sie eine zeitliche Begrenzung von 60 Sekunden fest.
12 Um die Paketerfassung zu starten, wählen Sie OK.

Es dauert einige Minuten, bis die Paketerfassung beginnt. Während der Erfassung werden die Daten auf das Azure Storage-Konto oder in eine lokale Datei Ihrer VM übertragen. Eine Liste mit Paketerfassungen finden Sie auf der Portalseite für den Network Watcher. Wenn Sie die Protokolle auf Azure Storage übertragen, kann die Aufnahme auf das Speicherkonto erfolgen, und Sie können dort die Erfassungsdatei mit der Endung *.cap herunterladen. Sie können dann die Paketerfassung, wie in Abschnitt 12.3.3 erwähnt, in einem Analyseprogramm öffnen. Die Netzwerkpaketerfassung aus Abbildung 12.5 weiter oben in diesem Kapitel stammt übrigens aus einer Paketerfassung mit Network Watcher.

12.4 Übung: Erstellen von Leistungswarnungen

Die Funktionen VM-Diagnose, Metriken und Network Watcher haben Ihnen hoffentlich vor Augen geführt, was mit Azure alles möglich ist, um Anwendungsprobleme zu beheben. Funktionen wie die Startdiagnose und die VM-Diagnose-Erweiterung sind besonders dann sinnvoll, wenn Sie diese genauso aktivieren und konfigurieren wie Sie VMs bereitstellen.

In dieser Übung konfigurieren Sie Metrikwarnungen, damit Sie sehen, welche Themen diese Warnhinweise haben können und wie sie aussehen, wenn sie bei Ihnen eingehen:

1 Gehen Sie im Azure Portal auf die in den vorausgegangenen Übungen erstellte VM.
2 Wählen Sie in der VM-Kategorie „Monitoring“ (Überwachung) die Option „Alerts“ (Warnhinweise).
3 Wählen Sie aus, eine Warnregel zu erstellen, und fügen Sie dann eine Bedingung hinzu, wenn der CPU-Anteil in den letzten 5 Minuten größer als durchschnittlich 10 % ist. Die neuesten Metriken sollten in einem Diagramm erscheinen. Passen Sie den Schwellenwert an, wenn bei 10 % keine Warnung ausgelöst wird.
4 Fügen Sie eine Aktionsgruppe hinzu, und geben Sie Ihr einen Namen und einen Kurznamen. Legen Sie für diese Übung beide Namen auf azuremol fest. Mit Aktionsgruppen können Sie wieder verwendbare Schritte definieren, wenn eine Warnung generiert wird, z. B. einer Reihe von Benutzer*innen eine E-Mail senden oder ein automatisiertes PowerShell-Skript oder eine Azure Logic-App ausführen.
5 Entdecken Sie die verfügbaren Aktionstypen, und wählen Sie dann E-Mail/SMS/Push/Voice aus.
6 Wählen Sie, wie Sie benachrichtigt werden möchten, z. B. per E-Mail oder Textnachricht. Für SMS- oder Sprachbenachrichtigungen können einige Anbieter Gebühren verlangen.
7 Wenn die Aktionsgruppe erstellt wurde, geben Sie der Warnung einen Namen, und legen Sie dann einen Schweregrad fest. Dieser Schweregrad ist nützlich, wenn Sie viele Warnungen definiert haben, die Sie bei der Selektierung und Priorisierung der Lösung unterstützen.
8 Erstellen Sie die Regel, wenn Sie bereit sind. Es dauert 10 bis 15 Minuten, bis die Regel aktiv wird und die von Ihnen definierten Benachrichtigungen generiert.

Dieses Beispiel ist ein grundlegend, also denken Sie an alle vorhandenen Warnungen und Benachrichtigungen für Anwendungen und Dienste und wie Sie diese Funktion bei der Ausführung von Workloads in Azure verwenden können.

Teil 3:

Sicherheit als Standard

Bedrohungen durch digitale Angriffe sind in einer vernetzten Welt, in der Anwendungen normalerweise rund um die Uhr mit dem Internet verbunden sind, allgegenwärtig. Diese Angriffe kosten Zeit, Geld und Kundenvertrauen. Ein zentraler Bestandteil der Erstellung hochredundanter und verteilter Anwendungen besteht darin, die Anwendungen und Ihre Daten zu schützen. Azure verfügt über mehrere integrierte Funktionen zum Schutz Ihrer Daten, einschließlich Verschlüsselung, Überwachung, digitaler Schlüsseltresor und Sicherung. In diesem Teil des Buchs erfahren Sie, wie Sie Ihre Anwendungen von Anfang an sichern und schützen können.

13 Sicherung, Wiederherstellung und Replikation

In den nächsten Kapiteln stellen wir Ihnen einige der Kernfunktionen und -dienste von Azure vor, die es Ihnen ermöglichen, Sicherheit in Ihre Anwendungen einzubauen. Das ist wahrscheinlich zu subjektiv gedacht: Sicherheit sollte kein Add-on-Feature oder eine Option sein. Sicherheit sollte vielmehr von Anfang voll in den Kern Ihrer Anwendung integriert werden. Dieses Kapitel ist eine Einführung in das Thema Sicherheit bei Azure und die Methoden der Sicherung und Wiederherstellung Ihrer Daten. Datensicherung ist auf den ersten Blick unbedingt ein zentrales Element beim Thema Sicherheit. Doch Sicherheit ist mehr als Datenverschlüsselung oder SSL-Zertifikate auf der Website. Wie steht es jedoch um den Schutz Ihrer Daten vor Ausfällen, Datenverlust und Hackerangriffen? Sicherungen und Replikation sind auch eine gute Überleitung vom Thema des Kapitels über hohe Verfügbarkeit und diesem Kapitel.

Sicherungen sind auf den ersten Blick eine triviale Aufgabe. Und als ehemaliger Sicherungsadministrator kann ich Ihnen sagen, dass es nicht viel Spannendes am Erstellen von Sicherungen und Rotationen gibt! Aber nur mit zeitnahen, funktionierenden Backups können Sie Ihre Anwendungen schützen und sicherzustellen, dass Sie Ihre Daten im Notfall schnell und zuverlässig wiederherstellen können. Sie können Ihre VMs auch von einer Azure-Region zur anderen replizieren. Diese Fähigkeit gründet auf den hochverfügbaren Konzepten, die wir bereits in Kapitel 7 behandelt haben.

In diesem Kapitel erfahren Sie, wie Sie VMs sichern und wiederherstellen können und wie Sie VMs automatisch über Azure replizieren. Diese Sicherungen und Wiederherstellungspunkte sind zum Schutz Ihrer Daten verschlüsselt.

13.1 Azure Backup

Ein großer Vorteil von Azure Backup besteht darin, dass es sich einerseits um einen Dienst handelt und andererseits um einen umfassenden Speicher für die

eigentlichen Sicherungen. Azure Backup kann VMs in Azure, On-Premises-VMs sowie physische Server und sogar VMs bei anderen Anbietern wie Amazon Web Services (AWS) schützen. Sie können Ihre Datensicherungen auf Ihren eigenen Speicherarrays On-Premises oder in einem Azure Recovery Services-Tresor speichern. Abbildung 13.1 zeigt, wie der Azure Backup-Dienst alle Ihre Sicherungsanforderungen schützen und organisieren kann.

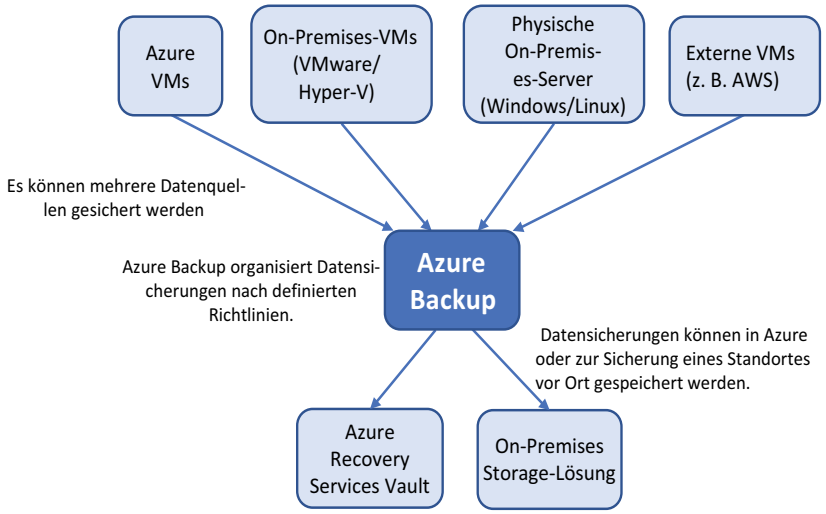


Abbildung 13.1 Über den zentralen Orchestrierungsdienst können Sie Daten von mehreren VMs oder physischen Server von verschiedenen Anbietern und an verschiedenen Standorten sichern. Azure Backup verwenden definierte Richtlinien, um Daten in einer bestimmten Häufigkeit oder einem bestimmten Zeitplan zu sichern. Sie können diese Datensicherungen anschließend in Azure oder einer On-Premises-Speicherlösung speichern. Eine durchgängige Datenverschlüsselung gewährleistet zusätzliche Sicherheit.

Im Kern verwaltet Azure Backup die Sicherungspläne und Datenaufbewahrung. Es organisiert die Datensicherung und -wiederherstellung. Für eine Datensicherung der Azure VMs müssen Sie keine Komponenten oder Agenten manuell installieren. Alle Sicherungs- und Wiederherstellungsvorgänge sind mit der Azure-Plattform integriert.

Das Backup von VMs an Ihrem Standort, auf physikalischen Servern oder bei anderen Anbietern wie AWS erfordert die Installation eines kleinen Agenten, der eine in beiden Richtungen sichere Kommunikation mit Azure ermöglicht. Diese sichere Kommunikation sorgt für eine durchgängig verschlüsselte Datenübertragung.

Die Backups der auf Azure gespeicherten Daten sind mit einem Schlüssel verschlüsselt, den Sie erstellen. Nur Sie haben Zugriff auf diese verschlüsselten Backups. Auch für verschlüsselte VMs ist ein Backup möglich. Dieses Thema behandeln wir im nächsten Kapitel, um sicherzustellen dass Ihre Sicherungen wirklich sicher sind.

Für den Netzwerkverkehrsfluss zur Datensicherung oder -wiederherstellung werden keine Gebühren erhoben. Sie zahlen nur für jede geschützte Instanz und abhängig von dem Speicher, den Sie auf Azure verbrauchen. Die Kosten für Azure Backup sind minimal, wenn Sie einen On-Premises-Speicherort verwenden, da dann keine Kosten für Azure Storage oder den Netzwerkverkehr anfallen.

13.1.1 Richtlinien und Datenaufbewahrung

Azure Backup verwendet ein inkrementelles Backup-Modell. Wenn Sie eine Instanz schützen, erfolgt beim ersten Sicherungsvorgang eine vollständige Datensicherung. Anschließend wird bei jedem Sicherungsvorgang eine inkrementelle Datensicherung vorgenommen. Jede dieser Sicherungen wird als *Wiederherstellungspunkt* bezeichnet. Inkrementelle Sicherungen sind eine zeitsparende Methode für eine optimale Nutzung von Speicherplatz und Netzwerkbandbreite. Nur Daten, die sich seit der letzten Sicherung veränderten haben, werden sicher an den Zielspeicherort für die Sicherung übertragen. Die Abbildung 13.2 verdeutlicht, wie eine inkrementelle Sicherung funktioniert.

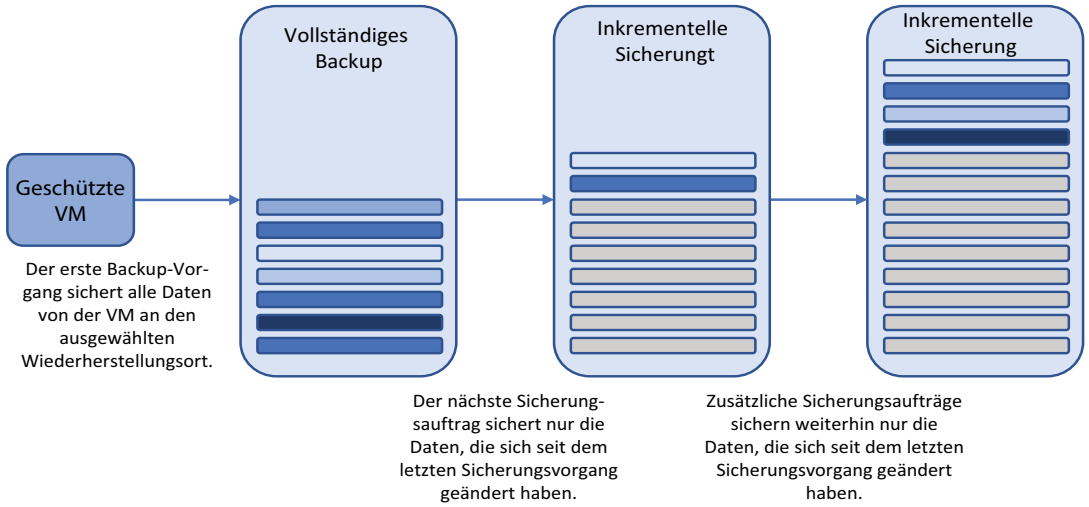


Abbildung 13.2 Eine inkrementelle Sicherung sichert nur die Daten, die sich seit dem letzten Sicherungsvorgang geändert haben. Die erste Sicherung ist immer ein komplettes Backup. Jeder nachfolgende Sicherungsauftrag sichert nur die Daten, die sich seit dem letzten Sicherungsvorgang geändert haben. Sie legen Richtlinien für die Häufigkeit von Voll-Backups fest. Mit dieser Methode minimieren Sie die Datenmenge, die sicher über das Netzwerk übertragen und am Zielspeicherort abgelegt wird. Azure Backup setzt alle inkrementellen Backups in eine Relation zueinander, um sicherzustellen, dass die Daten bei der Wiederherstellung konsistent und vollständig sind.

Mit Azure Backup können Sie für jede geschützte Instanz bis zu 9.999 Wiederherstellungspunkte speichern. Wenn Sie in regelmäßigen Abständen eine tägliche Sicherung durchführen, haben Sie für über 27 Jahre ausgesorgt. Wöchentliche Backups könnten Sie fast 200 Jahre lang durchführen. Damit würden Sie voraussichtlich die meisten Prüfsituationen abdecken! Sie können Backups auf täglicher, wöchentlicher, monatlicher oder jährlicher Basis aufbewahren, was in der Regel den meisten vorhandenen Backup-Richtlinien entspricht.

Um die optimale Backup-Strategie für Ihren Workload zu implementieren, müssen Sie Ihr akzeptables *Recovery Point Objective* (RPO) und das *Recovery Time Objective* (RTO) verstehen und festlegen.

RECOVERY POINT OBJECTIVE

Das RPO definiert den Punkt, bis zu dem Ihr neuestes Backup eine Wiederherstellung zulässt. Standardmäßig erstellt Azure Backup eine tägliche Sicherung. Sie definieren dann in den Datenaufbewahrungsrichtlinien, wie viele Tage, Wochen, Monate oder Jahre Sie diese Wiederherstellungspunkte behalten möchten. Obwohl die RPO in der Regel verwendet wird, um die maximale Größe an akzeptablen Datenverlusten zu definieren, sollten Sie auch überlegen, bis wohin die Sicherung zurückreichen soll. Die Abbildung 13.3 verdeutlicht, wie das RPO die Größe des akzeptablen Datenverlustes definiert.

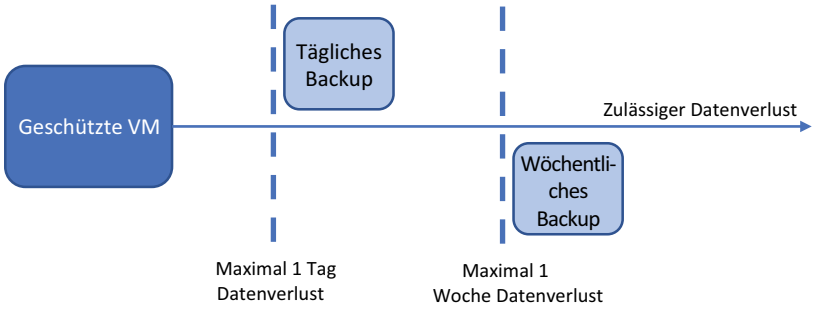


Abbildung 13.3 Das RPO legt fest, wie viel Datenverlust Sie für eine geschützte Instanz in Kauf nehmen können. Je länger die Dauer des RPO, desto größer ist der akzeptable Datenverlust. Ein RPO von einem Tag bedeutet, dass bis zu 24 Stunden Daten verloren gehen könnten, je nachdem, wann der Datenverlust im Verhältnis zur letzten Sicherung aufgetreten ist. Ein RPO von einer Woche bedeutet, dass Daten von bis zu sieben Tagen verloren gehen könnten.

Es kommt nur selten zu großen Ausfällen und großen Datenverlusten. Häufiger sind Vorfälle mit geringem Datenverlust oder eine Überschreibung von Daten. Diese Vorfälle werden oft erst bemerkt oder gemeldet, wenn der Datenverlust bereits eingetreten ist. An dieser Stelle ist die Datenaufbewahrungsrichtlinie für Ihre geschützten Instanzen von Bedeutung. Bei einer Richtlinie mit kurzer Aufbewahrungsdauer können Sie Daten möglicherweise nicht ab dem gewünschten Zeitpunkt wiederherstellen. Es bedarf eines ausgewogenen Verhältnisses zwischen der Aufbewahrung mehrerer Wiederherstellungspunkte und den Speicherkosten für die Aufbewahrung aller Wiederherstellungspunkte.

Azure Storage ist recht kostengünstig. Es kostet in der Regel weniger als 0,02 US-Dollar pro Gigabyte. Das entspricht etwa 2 US-Dollar pro Monat für eine 100 GB VM-Datensicherung (plus einer Gebühr für den eigentlichen Azure Backup-Dienst). Je nachdem, wie stark sich Ihre Daten ändern, kann sich die Größe inkrementeller Wiederherstellungspunkte schnell erhöhen. Die Aufbewahrung von Wiederherstellungspunkten über Wochen und Monate kann monatliche Kosten pro Instanz im zweistelligen Bereich mit sich bringen. Lassen Sie sich davon nicht entmutigen, aber planen Sie Ihre Anforderungen sorgfältig und beachten Sie die entstehenden Kosten. Der Preis von 0,02 US-Dollar pro Gigabyte für Azure Storage ist solange günstig, bis Sie pro geschützter Instanz Hunderte Gigabyte benötigen und Dutzende oder sogar Hunderte Instanzen schützen müssen.

Ich war einmal als Backup-Administrator tätig. Die Speicherkapazität wurde häufig zu einem zentralen Faktor, wenn ich feststellte, wie viele Wiederherstellungspunkte wir aufzubewahren hatten. Speicherkapazität und RPOs mussten häufig Kompromisse

eingehen. Wenn Sie anstatt einer On-Premises-Speicherlösung Azure Storage verwenden, brauchen Sie sich keine Sorgen um die verfügbare Speicherkapazität zu machen. Ich kann Ihnen garantieren, die Speicherkapazität ist höher als Ihr Kreditkartenlimit!

Recovery Time Objective

Das RTO bestimmt, wie schnell Sie Ihre Daten wiederherstellen können. Wenn Sie sich dafür entscheiden, Azure VMs zu sichern und die Wiederherstellungspunkte in einer On-Premises-Speicherlösung speichern, dauert Datenwiederherstellung sehr viel länger als bei einer Speicherung in Azure Storage. Das Gegenteil wäre der Fall, wenn Sie On-Premises-VMs oder physische Server in Azure Storage sichern würden. Abbildung 13.4 skizziert ein RTO.

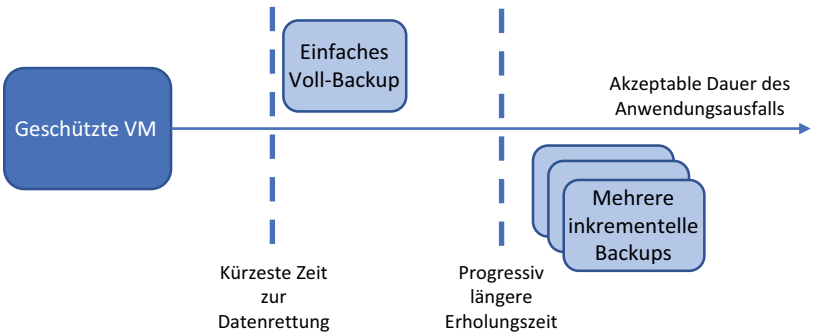


Abbildung 13.4 Das RTO legt fest, wie lange ein Datenwiederherstellungsprozess dauern und wie lange die Anwendung nicht verfügbar sein darf. Je mehr Wiederherstellungspunkte an der Wiederherstellung beteiligt sind, umso länger dauert das RTO. In ähnlicher Weise gilt: Je näher sich der Backup-Speicher an dem Wiederherstellungspunkt befindet, desto kürzer ist das RTO.

In beiden Fällen müssen die wiederherzustellenden Daten vom Speicherort des Wiederherstellungspunktes an den Wiederherstellungsort übertragen werden. Bei großen Wiederherstellungsvorgängen mit möglicherweise Hunderten Gigabyte zu übertragender Daten müssen, hängt es von der Netzwerkbandbreite ab, wie schnell Sie Anwendungen wieder verfügbar machen können.

Dasselbe gilt für langfristige Datenaufbewahrungsrichtlinien mit vielen aufeinanderfolgenden inkrementellen Wiederherstellungspunkten. Bei der Wiederherstellung der Daten müssen möglicherweise mehrere Wiederherstellungspunkte bereitgestellt und wiederhergestellt werden. Ihre Aufgabe ist es, zu bestimmen, wie weit Sie in der Zeit zurückgehen müssen und wie viel Zeit Sie brauchen, um die Daten wiederherzustellen.

Sowohl RPO als auch RTO variieren je nach Anwendung und Geschäftsgebrauch. Eine Anwendung, die Echtzeitaufträge verarbeitet, kann sich keine großen Ausfallzeiten leisten. Daher sind RPO und RTO wahrscheinlich sehr niedrig. Da Sie eine Datenbank in der Regel zum Speichern von Daten verwenden, werden Sie wahrscheinlich Toleranzen in der Anwendung entwickeln, anstatt sich auf Wiederherstellungspunkte zu verlassen. So auch bei Cosmos DB. Dort müssen Sie keine Backups erstellen. Die Azure-Plattform übernimmt die Replikation und den Schutz der Daten für Sie. Auf MySQL oder Microsoft SQL Server basierende benutzerdefinierte Lösungen

verwenden in der Regel ähnliche Cluster- und Replikationstypen. Sie gewährleisten, dass mehrere Kopien der Datenbank existieren. Daher ist nach dem Verlust einer Instanz keine Wiederherstellung eines Backups nötig. Datensicherungen dienen primär dazu, vor großen Ausfällen oder Datenkorruption zu schützen.

13.1.2 Backup-Pläne

Wie steuern Sie die Häufigkeit Ihrer Datensicherungen und die Aufbewahrung der Wiederherstellungspunkte? Bei Azure Backup sind diese Einstellungen in Richtlinien definiert. Sie erstellen diese Richtlinien, um Ihre verschiedenen Datenschutzszenarien abzudecken. Sie können sie für mehrere geschützte Instanzen wiederverwenden.

Definieren Sie zum Beispiel eine Sicherungsrichtlinie, die täglich um 18:30 Uhr eine Datensicherung erstellt. Sie möchten die täglichen Sicherungen sechs Monate aufbewahren und so rotieren, dass wöchentliche Sicherungen zwei Jahre erhalten bleiben. Geltende Bestimmungen schreiben vor, monatliche Sicherungen fünf Jahre lang aufzubewahren. Eine jährliche Datensicherung wird für zehn Jahre beibehalten. Diese gezeigten Aufbewahrungswerte wirken vielleicht etwas überzogen. Doch bei Anwendungen, die Kommunikation und Messaging enthalten, sind aus regulatorischen und Compliance-Gründen Datensicherungen häufig über diese langen Zeiträume aufzubewahren. Azure Backup ermöglicht eine flexible Definition von Richtlinien, die auf verschiedene Workloads von Anwendungen zugeschnitten sind und eine schnelle Umsetzung von Bestimmungen gewährleisten.

Jetzt testen

Sämtliche Azure-Backups werden in einem Recovery Services-Tresor gespeichert. Gehen Sie folgendermaßen vor, um eine Tresor- und Wiederherstellungsrichtlinie zu erstellen:

1. Öffnen Sie das Azure Portal, und klicken Sie im Menü oben links auf „Create a Resource“ (Ressource erstellen).
2. Suchen und wählen Sie Backup und Site Recovery, und klicken Sie dann auf „Create“ (Erstellen).
3. Erstellen Sie eine neue Ressourcengruppe, `azuremolchapter13`, und geben Sie dann einen Namen für den Tresor an, z. B. `azuremol`.
4. Wählen Sie einen Standort aus, und überprüfen und erstellen Sie dann den Tresor.
5. Wenn der Tresor erstellt wurde, wählen Sie Ressourcengruppen aus dem Menü links im Portal aus und, klicken Sie dann auf die von Ihnen erstellte Ressourcengruppe.
6. Wählen Sie aus der Liste verfügbarer Ressourcen Ihren Recovery Services-Tresor und anschließend aus dem Menü links „Backup Policies“ (Sicherungsrichtlinien), und fügen Sie eine Richtlinie hinzu.
7. Wählen Sie den Typ der Azure VM-Richtlinie, und geben Sie einen Namen für Ihre neue Richtlinie ein, wie zum Beispiel `molpolicy`. Standardmäßig wird jeden Tag ein Backup erstellt.
8. Wählen Sie aus dem Dropdown-Menü die am besten geeignete Zeitzone aus. Standardmäßig verwendet die Azure Universal Coordinated Time (UTC).

   Passen Sie bei Bedarf die Aufbewahrungsrichtlinien für tägliche, wöchentliche, monatliche und jährliche Backups an. Der Abschnitt zu den Konzepten der Sicherungs- und Aufbewahrungspläne beschreibt, wie Sie diese Werte auswählen. Diese Werte variieren in der Regel beim Erstellen und Anwenden von Sicherungsrichtlinien zum Schutz Ihrer VM-Instanzen.
9. Wählen Sie „Create“ (Erstellen), wenn Sie bereit sind.

Das einfache Leben

Wenn Sie im Azure Portal eine VM erstellen, können Sie auch VM-Backups konfigurieren. Sie können Azure Backup auf der Seite Einstellungen aktivieren, wenn Sie die Optionen für virtuelle Netzwerkeinstellungen, Diagnosen oder Problembehandlung konfigurieren. Sie können einen bestehenden Recovery Services-Tresor auswählen und dann eine Sicherungsrichtlinie erstellen oder verwenden. Es ist momentan nicht möglich, im Rahmen der Bereitstellung einer VM Backups in der Azure CLI oder Azure PowerShell zu aktivieren. Allerdings ist nach der Bereitstellung in der Regel nur der Befehl „postdeployment“ dazu nötig.

Ich halte viel davon, Sicherungsstrategien, Aufbewahrungsrichtlinien und Pläne zu erstellen. Daher haben wir in diesen Übungen zunächst einen Recovery Services-Tresor und entsprechende Richtlinien erstellt. Wer aber schnell eine VM erstellen und Backups ermöglichen möchte, kann das im Azure Portal in einem Schritt tun.

Sie verfügen nun über eine Sicherungsrichtlinie, in der auch Wiederaufbewahrungsrichtlinien für verschiedene Zeiträume festgelegt sind. Doch es gibt noch keine zu sichernden Daten. Nun erstellen wir mit der Cloud Shell eine VM, damit Sie eine Sicherung erstellen und in einer späteren Übung die Daten replizieren können.

Jetzt testen

Gehen Sie folgendermaßen vor, um eine Test-VM für das Backup und Replikation von Daten zu erstellen:

1. Wählen Sie oben im Azure Portal das Cloud Shell-Symbol aus.
2. Erstellen Sie eine VM mit `az vm create`, geben Sie den in der vorherigen Übung erstellten Ressourcengruppennamen ein, zum Beispiel `azuremolchapter13`, sowie anschließend einen VM-Namen, zum Beispiel `molvm`:

```
az vm create \
  --resource-group azuremolchapter13 \
  --name molvm \
  --image win2019datacenter \
  --admin-username azuremol \
  --admin-password P@ssw0rdMoL123
```

Eine Sicherungsrichtlinie wird definiert, und Ihre Test-VM ist fertig. Wenden Sie Ihre Sicherungsrichtlinie auf die VM an, um Azure Backup in Aktion zu erleben.

Jetzt testen

Gehen Sie folgendermaßen vor, um eine VM mit Ihrer definierten Richtlinie zu sichern:

1. Wählen Sie aus dem Menü links im Azure Portal „Resource Groups“ (Ressourcengruppen) aus.
2. Wählen Sie die Ressourcengruppe und dann die von Ihnen erstellte VM aus.
3. Gehen Sie unter „Operations“ (Vorgänge) auf Backup.
4. Achten Sie darauf, dass Ihr Recovery Services-Tresor ausgewählt ist, und wählen Sie dann aus dem Dropdown-Menü Ihre Sicherheitsrichtlinie aus.
5. Gehen Sie den Sicherungsplan und die Aufbewahrungsoptionen durch, und aktivieren Sie dann die Sicherung (das Backup). Die Anwendung der Sicherheitsrichtlinie dauert einige Sekunden.

6 Wenn die Richtlinie aktiviert ist, kehren Sie zu den Sicherungseinstellungen zurück. Der VM-Status meldet „`Warnung (erste Sicherung steht aus)`".

7 Wählen Sie die Schaltfläche „Backup Now" (Jetzt sichern) aus, um die erste Sicherung zu erstellen (siehe Abbildung 13.5).

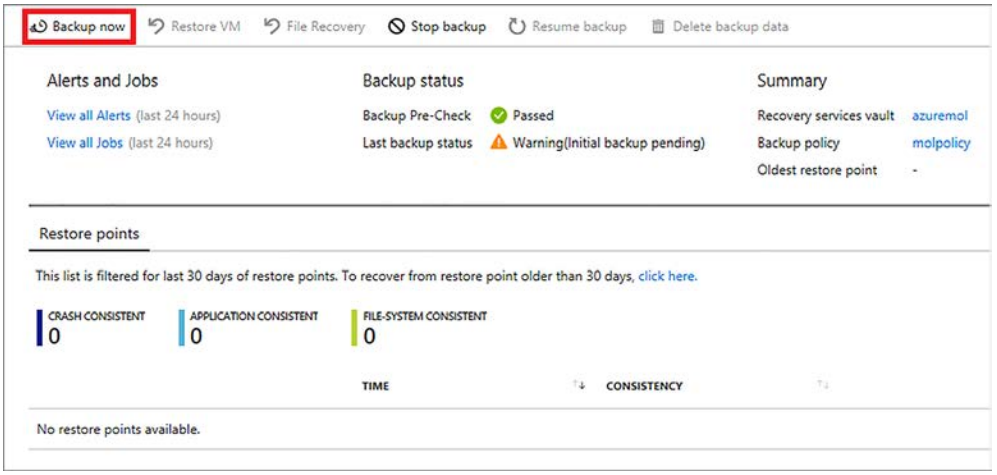


Abbildung 13.5 Wählen Sie die Schaltfläche „Backup Now" (Jetzt sichern) aus, um die erste Sicherung zu erstellen. Der Status aktualisiert sich nach Abschluss der Sicherung mit Informationen über die Zeit der letzten Sicherung sowie den letzten und den ältesten Wiederherstellungspunkt.

Bis die erste Sicherung komplett abgeschlossen ist, können 15 bis 20 Minuten vergehen. Um den Fortschritt der Sicherung zu überwachen, klicken Sie auf „View All Jobs" (Alle Aufträge anzeigen). Es gibt weder einen Fortschrittsbalken noch eine Prozentanzeige, aber Sie können sich vergewissern, ob der Auftrag noch läuft.

Mehr brauchen Sie nicht, um VMs zu sichern und Ihre Daten auf Azure zu schützen! Lesen Sie weiter, um zu erfahren, wie Sie die Daten bei Bedarf wiederherstellen können.

13.1.3 Wiederherstellen einer VM

Mit Azure Backup können Sie eine komplette VM oder eine Dateiebene wiederherstellen. In all den Jahren, in denen ich in diesem Bereich tätig war, waren Wiederherstellungsvorgänge auf Dateiebene die häufigere der beiden Varianten. Diese Art von Wiederherstellungsauftrag wird in der Regel durchgeführt, wenn Dateien gelöscht oder versehentlich überschrieben worden sind. Wiederherstellungen auf Dateiebene bestimmen in der Regel die Aufbewahrungsrichtlinien für Ihre Sicherungen. Je wichtiger die Daten sind, desto wahrscheinlicher ist es, dass Sie Sicherungen länger aufbewahren wollen, etwa für den Fall, dass Sie mitten in der Nacht einen Anruf erhalten und gebeten werden, eine Datei von vor sechs Monaten wiederherzustellen.

Eine komplette VM-Wiederherstellung stellt wie zu erwarten die gesamte VM wieder her. Ich habe äußerst selten eine komplette VM wiederhergestellt, um eine gelöschte VM wieder online zu bringen. Ein sehr gutes Beispiel für eine komplette VM-Wiederherstellung ist die Bereitstellung einer VM zu Testzwecken, die in ihrer Funktion dem Original entspricht. Sie können eine VM wiederherstellen und damit anschließend ein Software-Upgrade oder ein anderes Wartungsverfahren testen. So können Sie mögliche Probleme erkennen und einen Plan für die tatsächlich eingesetzte VM erstellen.

Außerdem ist es wichtig, Ihre Sicherungen regelmäßig zu testen. Warten Sie nicht, bis eine Situation eintritt, in der Sie Daten in einem realen Szenario wiederherstellen müssen. Verlassen Sie sich auf Azure Backup, aber achten Sie darauf, wie und wo Sie die Daten bei Bedarf wiederherstellen können!

Wiederherstellung auf Dateiebene

Eine Wiederherstellung auf Dateiebene ist bei Azure Backup eine ziemlich geniale Angelegenheit. Azure erstellt ein Wiederherstellungsskript, dass Sie herunterladen und ausführen können. Es ermöglicht Ihnen eine flexible Herstellung von Dateien. Dieses Wiederherstellungsskript ist kennwortgeschützt, damit Sie nur den Wiederherstellungsprozess ausführen können. Wenn Sie das Wiederherstellungsskript ausführen, werden Sie aufgefordert, das Kennwort für den weiteren Verlauf einzugeben. Das Fenster zum Herunterladen des Wiederherstellungsskripts ist in Abbildung 13.6 dargestellt.

Beim Ausführen des Wiederherstellungsskripts wird Ihr Wiederherstellungspunkt als lokales Dateisystem auf Ihrem Computer verbunden. Bei Windows VMs wird ein PowerShell-Skript generiert und ein lokales Volumen angeschlossen, beispielsweise F:. Bei Linux-VMs wird der Wiederherstellungspunkt als Datenträger in Ihrem Home Volume bereitgestellt, beispielsweise /dev/sdc1. In beiden Fällen zeigt das Wiederherstellungsskript deutlich an, wo Sie Ihre Dateien finden können.

Kehren Sie nach Abschluss der Wiederherstellung der Dateien aus dem Recovery Vault zum Azure Portal zurück, und wählen Sie „Unmount Disks“ (Bereitstellung der Datenträger aufheben). Dieser Prozess löst die Datenträger von Ihrem lokalen Computer und gibt sie für den Einsatz im Recovery Vault zurück. Machen Sie sich keine Sorgen, wenn Sie im Eifer des Gefechts vergessen, die Bereitstellung aufzuheben, weil Sie Dateien für eine zur Produktion eingesetzte VM schnell wiederherstellen müssen! Azure trennt die angeschlossenen Wiederherstellungspunkte automatisch nach 12 Stunden.

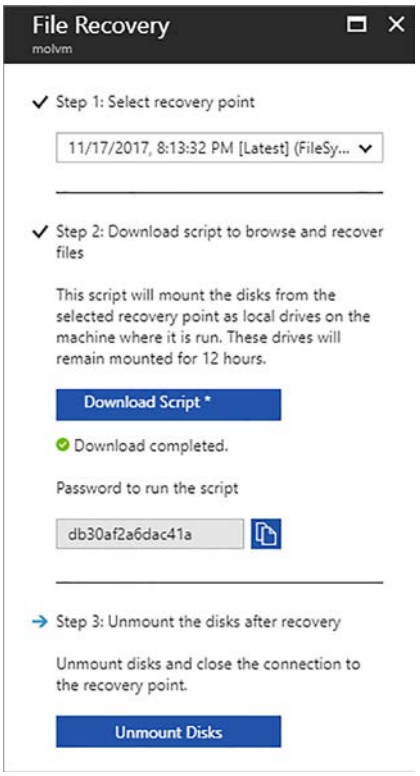


Abbildung 13.6 Bei Wiederherstellung auf Dateiebene wählen Sie den entsprechenden Wiederherstellungspunkt aus. Dann wird ein Wiederherstellungsskript auf Ihren Computer heruntergeladen. Sie können dieses Skript nur ausführen, indem Sie das generierte Kennwort eingeben. Das Wiederherstellungsskript stellt den Wiederherstellungspunkt als lokales Volumen auf Ihrem Computer bereit. Trennen Sie nach dem Wiederherstellen der benötigten Dateien die Datenträger von Ihrem Computer. Dadurch können Sie wieder im Recovery Vault verwendet werden.

Komplette VM-Wiederherstellung

Eine vollständige VM-Wiederherstellung erstellt eine VM, verbindet diese mit dem virtuellen Netzwerk und schließt alle virtuellen Datenträger an. So sieht der Prozess für eine vollständige VM-Wiederherstellung aus. Diese Wiederherstellungsübung ist eine gute Erfahrung, da es sich immer empfiehlt, Wartungsupdates vor dem tatsächlichen Einsatz zu testen.

Jetzt testen

Gehen Sie folgendermaßen vor, um eine VM vollständig wiederherzustellen:

1. Wählen Sie aus Ihrer Ressourcengruppe die VM, die Sie bei der vorherigen Übung wiederhergestellt haben.
2. Wählen Sie aus dem Menü links in der VM die Sicherungsoption aus. Die Backup-Übersicht sollte einen Hinweis geben, dass ein Wiederherstellungspunkt erstellt wurde (siehe Abbildung 13.7). Sollte dies nicht der Fall sein, warten Sie einige Minuten, und führen Sie diese Übung dann fort. Oder lesen Sie sich einfach durch, was der Prozess beinhaltet.

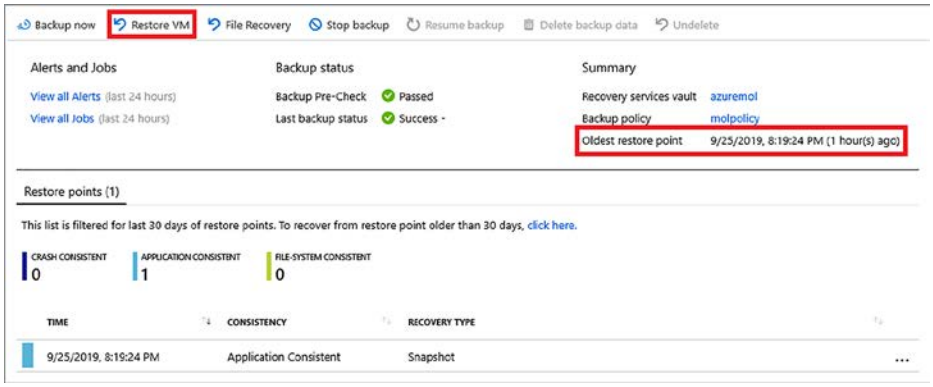


Abbildung 13.7 Nach Abschluss der VM-Sicherung sind auf der Übersichtsseite die Daten der letzten Sicherung und die verfügbaren Wiederherstellungspunkte angegeben. Wählen Sie Wiederherstellen einer VM, um den Wiederherstellungsprozess zu starten.

3. Wählen Sie die Schaltfläche „Restore VM“ (VM wiederherstellen), wählen Sie einen Wiederherstellungspunkt aus der Liste aus, und wählen Sie dann OK aus.
4. Wählen Sie einen Wiederherstellungspunkt, und wählen Sie aus, wie die VM wiederhergestellt werden soll. Sie können eine neue VM erstellen oder eine vorhandene VM ersetzen.

   Die Standardoption besteht darin, eine neue VM zu erstellen. In dieser Konfiguration wird eine neue VM erstellt und mit dem angegebenen virtuellen Netzwerkverbunden,unddieDatenträgerwerdenwiederhergestelltundverbunden.

   Sie können auch eine vorhandene VM ersetzen. In diesem Szenario werden die Datenträger aus der Sicherung wiederhergestellt und an die vorhandene VM angefügt. Jedes virtuelle Netzwerk oder andere Konfigurationsoptionen, die auf die VM angewendet werden, werden beibehalten.
5. Wählen Sie für diese Übung die Wiederherstellung auf einer neuen VM aus. Geben Sie einen Namen für die wiederhergestellte VM ein, beispielsweise `restoredvm`. Überprüfen Sie anschließend die Einstellungen für das virtuelle Netzwerk und den Storage. In der Produktion wird die wiederhergestellte VM in der Regel mit einem separaten virtuellen Netzwerk verbunden, damit der Datenverkehr der Produktion nicht beeinträchtigt ist.
6. Gehen Sie auf OK und anschließend auf „Restore“ (Wiederherstellen).

Es dauert einige Minuten, um den Wiederherstellungspunkt zu verbinden und eine wiederhergestellte VM mit den zuvor verbundenen Datenträgern zu erstellen. An dieser Stelle können Sie sich mit der wiederhergestellten VM verbinden, um Software-Upgrades zu testen oder große Datenmengen nach Bedarf wiederherzustellen.

Sie können auch eine Web App sichern. Diese Methode ist also nicht nur für VMs bestimmt. Der Prozess ist ein wenig anders, aber die Begriffe sind die gleichen. Wenn Sie Ihr Anwendungsmodell wie eine Web App auf eine PaaS-Lösung umstellen, bedeutet dass nicht, dass Sie die Grundlagen der Datensicherung und -aufbewahrung vergessen können!

13.2 Azure Site Recovery

Erinnern Sie sich an das Thema Cosmos DB? Sie erfuhren, dass es möglich ist, Ihre Daten aus Gründen der Redundanz und Fehlertoleranz per Knopfdruck auf eine ganz anderen Azure-Region zu replizieren. Dies ist auch mit ganzen VMs möglich. Azure Site Recovery ist ein leistungsstarker Service, der weit mehr kann, als VMs in eine andere Region zu replizieren. Abbildung 13.8 verdeutlicht, wie Sie mit Azure Site Recovery die Workloads zwischen Standorten orchestrieren können.

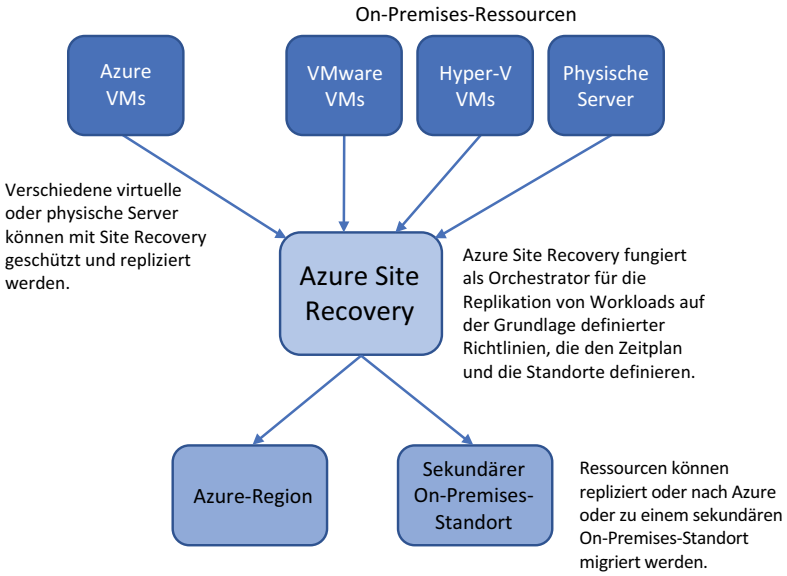


Abbildung 13.8 Azure Site Recovery orchestriert die Replikation und Migration physischer oder virtueller Ressourcen an einen anderen Ort. Sie können sowohl On-Premises-Standorte als auch Azure als Quell- und Zielpunkte für den Schutz, die Replikation oder Migration verwenden.

Sie können Azure Site Recovery nicht nur für Azure VMs verwenden. Es ist auch möglich, On-Premises VMware oder Hyper-V MVs zur Notfallwiederherstellung oder im Rahmen einer Migration zu Azure zu replizieren. Sie können Azure Site Recovery überdies auch nur als Orchestrator verwenden, um On-Premises-VMs von einem Standort auf einen sekundären On-Premises-Standort zu replizieren.

So wie Azure Backup nicht „nur mit Azure“ arbeitet, repliziert Azure Site Recovery auch nicht „nur Azure VMs“. Sie können sowohl Azure Backup als auch Azure Site Recovery als Hybrid-Lösungen zur Sicherung und für die Notfallwiederherstellung Ihrer Daten nutzen. Sie können alle Ihre Workloads On-Premises oder in Azure mit diesen Azure-Diensten schützen. Anschließend können Sie eine einzige Berichtsstruktur für die Compliance und Validierung erstellen, um zu gewährleisten, dass alle Workloads, die Sie schützen möchten, auch wirklich vor Datenverlust geschützt sind.

Wozu würden Sie Azure Site Recovery nutzen? Am häufigsten wird es zur Replikation und Migration eingesetzt.

Die Replikation schützt Sie vor einem vollständigen Ausfall einer Azure-Region. Es wäre eine Katastrophe, wenn eine ganze Region offline ginge. Doch wer in der IT arbeitet weiß, dass alles möglich ist. Auch die in Kapitel 7 erwähnten Verfügbarkeitssets und Verfügbarkeitszonen schützen in der Regel nur vor einem kleineren Ausfall in einer Azure-Region. Fällt die gesamte Region aus, funktioniert auch Ihre App nicht mehr. Site Recovery repliziert Ihre gesamte Anwendungsumgebung, einschließlich virtueller Netzwerkressourcen, auf eine sekundäre Azure-Region. Sie können diesen sekundären Standort auf Knopfdruck online stellen und aktivieren. Der Datenverkehr wird dann auf diesen sekundären Standort umgeleitet, und Ihre Kund*innen können Ihr Angebot weiter nutzen. Abbildung 13.9 verdeutlicht im Überblick, wie Azure Site Recovery Ihre Umgebung schützt.

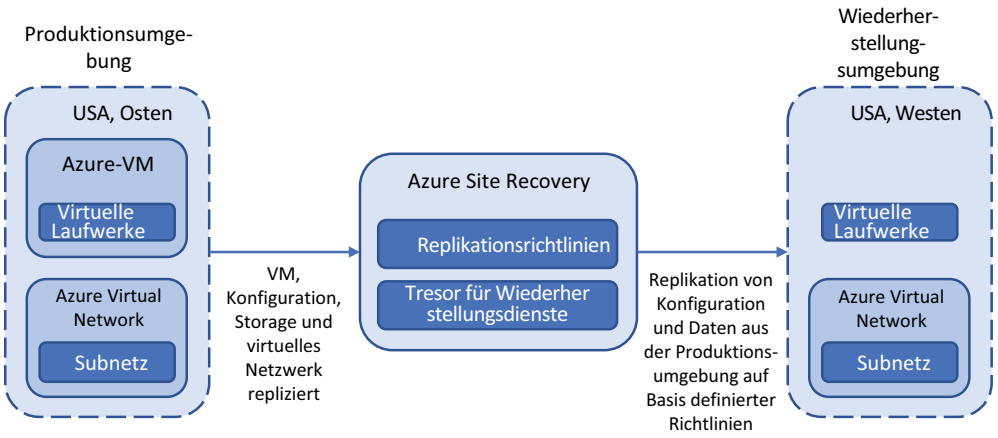


Abbildung 13.9 Azure Site Recovery repliziert Konfigurationen, Daten und virtuelle Netzwerke aus der Produktionsumgebung auf eine Wiederherstellungsumgebung. Die VMs werden erst nach dem Start eines Failovers in der Wiederherstellungsumgebung erstellt. Nur Daten werden repliziert.

Bei der VM handelt es sich lediglich um Metadaten, die bestimmen, wie groß die VM ist, welche Datenträger angeschlossen sind und mit welchen Netzwerkressourcen die VM verbunden ist. Diese Metadaten werden repliziert. Nach dem Start eines Failovers ist es dann möglich, die VMs schnell zu erstellen. Die virtuellen Datenträger werden auf die Wiederherstellungsumgebung repliziert und angeschlossen, wenn während eines Failover-Ereignisses eine wiederhergestellte VM erstellt wird.

Für die Replikation von Azure zu Azure gibt es keinen definierten Replikationsplan. Die Replikation der Datenträger erfolgt nahezu in Echtzeit. Ändern sich Daten auf den virtuellen Quelldatenträgern, werden diese auf die Wiederherstellungsumgebung kopiert. Für hybride Workloads, bei denen Sie VMware- oder Hyper-V-VMs lokal schützen, definieren Sie Richtlinien, die den Replikationsplan steuern.

Wie werden Daten bei einer Replikation von Azure auf Azure in Echtzeit repliziert? In der Produktionsumgebung wird ein Storage-Konto-Cache erstellt (siehe Abbildung 13.10). Auf die virtuellen Produktionsdatenträger geschriebene Daten werden sofort in diesen Storage-Konto-Cache repliziert. Anschließend wird der Storage-Konto-Cache auf die Wiederherstellungsumgebung repliziert. Dieser Storage-Konto-Cache dient als ein Puffer. Daher beeinträchtigt eine verzögerte Replikation an einem entfernten Wiederherstellungsort auch den Produktionsworkload nicht.

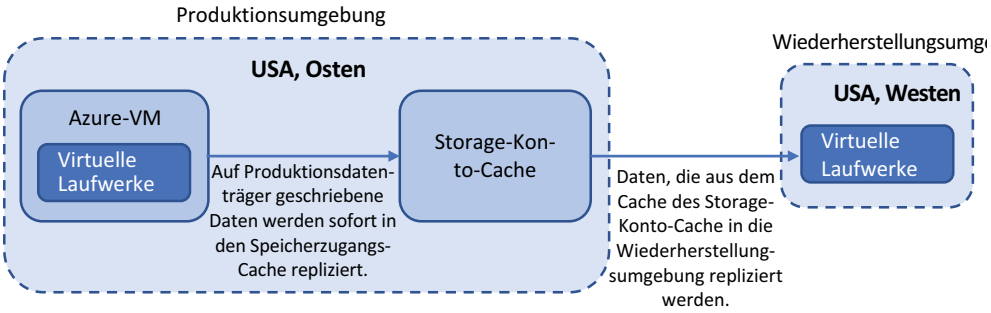


Abbildung 13.10 Änderungen auf dem Produktionsdatenträger werden sofort in einen Storage-Konto-Cache kopiert. Dieser Storage-Konto-Cache verhindert negative Auswirkungen auf die Produktionsworkloads, da Änderungen mit einer Verzögerung an den entfernten Wiederherstellungsort repliziert werden. Die Änderungen aus dem Storage-Konto-Cache werden dann auf den entfernten Wiederherstellungspunkt repliziert, um die Datenkonsistenz zu erhalten.

Eine Replikation von Azure auf Azure lässt sich einfach konfigurieren. Es dauert jedoch eine Weile, alle dafür notwendigen replizierten Ressourcen zu erstellen und die erste Datenreplikation durchzuführen. In der Übung am Ende des Kapitels konfigurieren Sie eine Replikation von Azure zu Azure.

Welchen Nutzen haben VMs, die mit Azure Site Recovery in eine sekundäre Region repliziert wurden? Ich drücke Ihnen die Daumen und hoffe, dass Sie diese VMs nicht brauchen werden! Doch es gibt einige Szenarien, in denen das unumgänglich ist.

Das erste Szenario ist eindeutig: bei einem schweren Ausfall. Steht eine Azure-Region überhaupt nicht mehr zur Verfügung, etwa weil sich dort eine Naturkatastrophe ereignet hat, können Sie einen Failover Ihrer Ressourcen starten. Dieses Failover erstellt Azure Site Recovery in der Wiederherstellungsregion anhand der replizierten VM-Metadaten die nötigen VMs und stellt die entsprechenden virtuellen Datenträger und Netzwerkverbindungen bereit. Sie können hier auch proaktiv handeln und bereits bei der Ankündigung einer Naturkatastrophe in einer Azure-Region einen Failover auslösen, noch *bevor* das Ereignis stattfindet. Auf diese Weise entscheiden Sie, wann es zu einer potenziellen Ausfallzeit kommen kann, bei der die Ressourcen in eine sekundäre Region überwechseln, was in der Regel außerhalb der üblichen

Geschäftsstunden geschieht. Sobald das vorausgesagte Ereignis in der primären Azure-Region vorbei ist, können Sie Ihre Ressourcen wieder dorthin umlenken und den Normalbetrieb wieder aufnehmen.

Im zweiten Szenario für einen möglichen Failover testen Sie, ob der Prozess funktioniert. So wie Sie Sicherungen regelmäßig testen sollten, sollten Sie auch einen Replikations- und Failover-Plan regelmäßig testen. Es dürfte recht peinlich und stressig für Sie werden, wenn Sie eine sekundäre Region online bringen müssen und feststellen, dass es einige Fehlkonfigurationen in den virtuellen Netzwerken gibt, oder der Failover einer Anwendungen nicht reibungslos gelingt. Glücklicherweise gibt es bei Azure eine spezielle Option, um Failover zu testen. In der Regel wird in der sekundären Region ein isoliertes virtuelles Azure-Netzwerk eingesetzt, die Produktionsworkloads laufen weiterhin über die primäre Region. Testen Sie den Failover-Prozess regelmäßig, wenn Sie Azure Site Recovery verwenden!

13.3 Übung: Konfigurieren einer VM für Site Recovery

Für die Konfiguration der Replikation von VMware oder Hyper-V, die On-Premises liegen, gibt es eine Reihe von Voraussetzungen. Dieses Feature eignet sich hervorragend für die Notfallwiederherstellung und zur Migration von VMs zu Azure. Doch dies dauert sehr viel länger als eine Mittagspause! Wenn Sie mehr über diese Szenarien erfahren möchten, gehen Sie auf http://mng.bz/x71V.

Richten wir nun mit der Test-VM, für die Sie weiter oben ein Backup erstellt haben, eine Replikation von Azure auf Azure ein:

1. Wählen Sie aus dem Menü links im Azure Portal die Ressourcengruppen aus.
2. Wählen Sie die Ressourcengruppe, die Sie in den vorherigen Übungen verwendet haben, wie beispielsweise azuremolchapter13.
3. Wählen Sie die VM, die Sie in den früheren Übungen erstellt haben, wie zum Beispiel molvm.
4. Wählen Sie im VM-Fenster in dem Menü links „Disaster Recovery“ (Notfallwiederherstellung) aus.
5. Unter den erweiterten Einstellungen sehen Sie sich die Standardeinstellungen an, die von Azure Site Recovery verwendet werden, um eine Ressourcengruppe und ein virtuelles Netzwerk am Zielstandort zu erstellen. Ein Storage-Konto-Cache für die Replikation der virtuellen Quell-Datenspeicher wird erstellt, ebenso ein Recovery Services-Tresor und eine Richtlinie zur Kontrolle des Replikationsvorgangs.
6. Sie brauchen an dieser Stelle keine Änderungen vorzunehmen. Falls Sie Site Recovery allerdings in der Produktion verwenden und mehrere VMs schützen müssen, sollten Sie prüfen, ob die VMs zu den bestehenden replizierten virtuellen Netzwerken und Subnetzen passen. Für diese Übung prüfen Sie die Replikation mit den Standardwerten, und aktivieren Sie sie.

Nun können Sie wieder an die Arbeit gehen. Im Ernst! Es dauert eine Weile, bis alle replizierten Ressourcen konfiguriert sind und die erste Datensynchronisierung abgeschlossen ist. Verschwenden Sie keine Zeit, es sei denn, Ihr Chef ist einverstanden damit, dass Sie Ihre Mittagspause heute gewaltig ausdehnen!

Schützen Sie Ihre Sicherungen (Backups) vor dem Löschen

Ich hoffe, dass Sie als bewährte Methode Ressourcengruppen und ihre Ressourcen am Ende jedes Kapitels gelöscht haben, um Ihre Azure-Gutschriften für die Verwendung im Rest des Buches zu nutzen.

Bei VMs, die mit Azure Backup oder Site Recovery geschützt sind, können Sie den Recovery Services-Tresor oder die Ressourcengruppe für die VM nicht löschen. Die Azure-Plattform weiß, dass Sie über gesicherte und replizierte aktive Daten verfügen und verhindert, dass diese Ressourcen gelöscht werden.

Wenn Sie geschützte VMs löschen möchten, müssen Sie zunächst aktive Sicherungsaufträge oder replizierte VMs deaktivieren. Anschließend können Sie die geschützten Daten behalten oder entfernen. Für die Übung in diesem Kapitel löschen Sie die Wiederherstellungspunkte. Als Sicherheitsfeature löscht Azure automatisch diese Wiederherstellungspunkte und ermöglicht Ihnen, die Löschung der letzten 14 Tage wieder aufzuheben. Hier können Sie nichts konfigurieren, und Sie können diese Soft-Deleted-Wiederherstellungspunkte nicht erzwungenermaßen entfernen. Ich empfehle es nicht, aber Sie können auch die Soft-Delete-Funktion eines Recovery Services-Tresors deaktivieren, indem Sie die Eigenschaften des Tresors im Azure Portal auswählen.

Die gute Nachricht ist, dass der Rest der Ressourcengruppe gelöscht werden kann, und Sie nicht für diese Soft-Deleted-Wiederherstellungspunkte zahlen. Wenn der 14-tägige Soft-Delete-Zeitpunkt vorbei ist, kann der Recovery Services-Tresor als normal gelöscht werden. Das Ziel besteht darin, Sie vor versehentlichen oder böswilligen Löschungen von Wiederherstellungspunkten zu schützen und Ihnen Zeit zu geben, um zu erkennen, ob sie tatsächlich benötigt werden, und sie wiederherzustellen.

14 Datenverschlüsselung

Die Sicherheit Ihrer Daten ist wichtig. Genauer gesagt ist die Sicherheit der Daten Ihrer Kund*innen von entscheidender Bedeutung. Es vergeht kaum eine Woche, ohne dass ein großes Unternehmen wegen einer Datenschutzverletzung in die Nachrichten kommt. Häufig werden diese Vorfälle durch mangelnde Sicherheit, Fehlkonfiguration oder schlichte Unachtsamkeit verursacht. In diesem digitalen Zeitalter stellt es für Angreifer keine besondere Herausforderung dar, ihre Angriffsversuche zu automatisieren, um sich Zugang zu Ihren Daten zu verschaffen. Die Zeit, um sich von einem Sicherheitsvorfall auf Anwendungsebene zu erholen, ist unbedeutend verglichen mit der Dauer bis das Unternehmen das *Vertrauen* seiner Kund*innen zurückgewonnen hat, nachdem ihre Daten von einem Angriff betroffen waren.

Azure enthält Verschlüsselungsfunktionen, die Ihnen das notwendige Know-how bieten, um Ihre Daten zu sichern und den Zeitaufwand dafür entsprechend reduzieren. In diesem Kapitel untersuchen wir die Verschlüsselung von Daten, die in Azure Storage, auf verwalteten Datenträgern oder der gesamten VM gespeichert sind. Ganze Bücher wurden zum Thema Datenverschlüsselung verfasst, und dieses Kapitel bietet keine tiefen Einblicke in Verschlüsselungsmethoden und Überlegungen. Stattdessen erfahren Sie darin, wie Sie einige der Kernfunktionen und -dienste von Azure einsetzen können, um Ihre Daten während des gesamten Anwendungszyklus zu sichern.

14.1 Was ist Datenverschlüsselung?

Überprüfen Sie bei einem Online-Einkauf, ob in der Adressleiste ein kleines Vorhängeschloss-Symbol zu sehen ist, der anzeigt, dass die Website HTTPS verwendet? Warum ist es suboptimal, Ihre Kreditkartendaten über eine reguläre, ungesicherte HTTP-Verbindung zu senden? Jedes Bit an Daten in einem Netzwerk-Paket, das zwischen den Geräten übertragen wird, könnte möglicherweise überwacht und untersucht werden. Abbildung 14.1 zeigt, warum der Online-Einkauf ohne HTTPS-Verbindung negative Folgen für Ihre Kreditkartenabrechnung haben könnte.

Für Webserver ist es unzulässig, unsichere Verbindungen zu verwenden. Jeder Web App, die Sie auf Azure erstellen, wird automatisch ein Wildcard-SSL-Zertifikat zugewiesen.

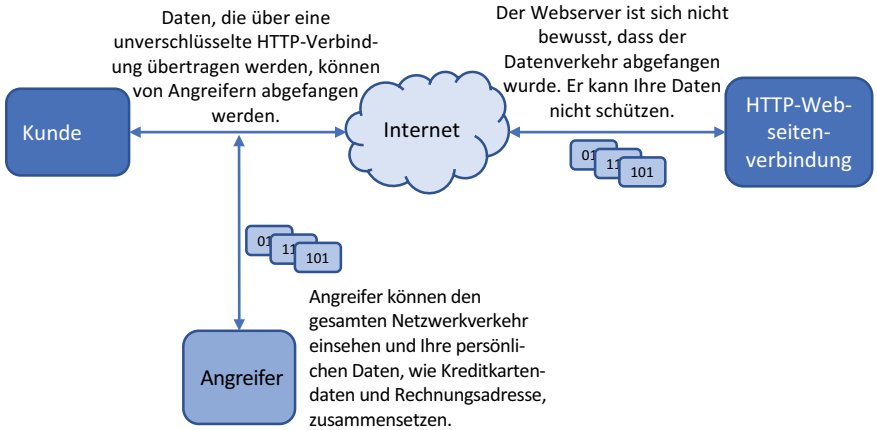


Abbildung 14.1 In diesem einfachen Beispiel könnte ein Angreifer den Netzwerkverkehr abfangen, der über eine unverschlüsselte HTTP-Verbindung übertragen wird. Da Ihre Daten nicht verschlüsselt sind, kann der Angreifer die Netzwerkpakete zusammenfügen und Ihre persönlichen und finanziellen Informationen erhalten. Wenn Sie sich stattdessen über eine verschlüsselte HTTPS-Verbindung mit dem Webserver verbinden, kann ein Angreifer den Inhalt der Netzwerkpakete nicht lesen und die Daten nicht sehen.

Ein *SSL-Zertifikat* ist eine digitale Komponente, um den Webserver zu sichern und es einem Webbrowser zu ermöglichen, die Verbindung zu validieren. Ein Wildcard-SSL-Zertifikat kann in der gesamten Domain verwendet werden, z. B. *.azurewebsites.net, die Standarddomain für Web Apps. Wenn Sie in Kapitel 3 eine Web App erstellt haben, könnten Sie Folgendes hinzugefügt haben: https:// to the web address and started to use encrypted communications with your web apps. Das ist der ganze Trick!

Benutzungsdefinierte SSL-Zertifikate sind relativ billig und einfach zu implementieren. Durch Projekte wie Let's Encrypt (https://letsencrypt.org) können Sie ein Zertifikat kostenlos erhalten und Ihren Webserver in wenigen Minuten automatisch konfigurieren. Sie können auch ein App Service-Zertifikat kaufen und verwenden, das direkt in Web Apps integriert ist. App Service-Zertifikate werden in Azure Key Vault gespeichert, das wir in Kapitel 15 näher betrachten werden.

Bei der Entwicklung von Anwendungen auf Azure sollten Sie stets möglichst sichere Kommunikationsverfahren implementieren. Mit diesem Ansatz können Sie Ihre Daten bei der Übertragung sichern, aber was passiert, wenn diese Daten auf dem Datenträger gespeichert werden? Ein ähnlicher Prozess existiert auch für Datenträger und VMs, der Ihre ruhenden Daten sichert und schützt. Abbildung 14.2 zeigt, wie Datenträger- und VM-Verschlüsselung funktioniert.

Ich hoffe, diese vereinfachten Beispiele zur Datenverschlüsselung in Azure motivieren Sie zur Implementierung der Verschlüsselung bei der Entwicklung von Anwendungen in Azure. Die meisten Kunden erwarten, dass ihre Daten geschützt werden. Viele Unternehmen müssen Vorschriften und Compliance-Vorschriften befolgen, die eine Verschlüsselung erfordern. Mögliche Bußgelder, die dem Unternehmen aufgrund einer Datenschutzverletzung oder des Verlustes des Kundenvertrauens entstehen, sind nicht die einzigen Konsequenzen, die Sie berücksichtigen sollten. Bedenken Sie dabei auch das Risiko, dass personenbezogene und finanzielle Kundendaten preisgegeben werden und welche Auswirkungen dies auf ihr tägliches Leben haben könnte.

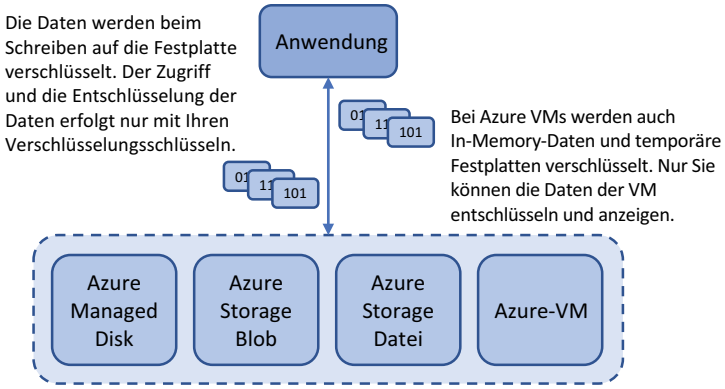


Abbildung 14.2 Bei der Verschlüsselung Ihrer Daten kann keine Person außer Ihnen den Inhalt entschlüsseln und anzeigen. Wenn ein Angreifer Zugriff auf einen virtuellen Datenträger oder einzelne Dateien erhält, kann er ihren Inhalt nicht entschlüsseln. Verschlüsselungsmethoden können kombiniert werden: Kund*innen können sich mit Ihrem Netzwerk über HTTPS verbinden. Sie können erzwingen, dass der Datenverkehr zu Storage-Konten über HTTPS erfolgt, und Sie können dann die Daten, die auf dem Datenträger gespeichert werden, verschlüsseln.

Der Gedanke daran, dass Ihre eigenen Daten offengelegt werden, gefällt Ihnen wahrscheinlich nicht, deshalb unternehmen Sie alles mögliche, um die Daten Ihrer Kund*innen zu schützen.

14.2 Verschlüsselung ruhender Daten

Die Verschlüsselung von Daten spielt also eine zentrale Rolle. Wie wird sie bei Azure umgesetzt? Machen Sie einfach damit weiter, was Sie in diesem Buch schon gelernt haben! Gleich zu Beginn habe ich erwähnt, dass alle Ihre VMs verwaltete Datenträger verwenden sollten, richtig? Dafür gibt es viele gute Gründe, und einer davon ist Sicherheit. Ein verwalteter Datenträger wird automatisch verschlüsselt. Sie müssen nichts konfigurieren, und zwar ohne Auswirkungen auf die Leistung bei der Aktivierung. Hier gibt es keine Alternative. Ihre ruhenden Daten werden auf verwalteten Datenträgern automatisch verschlüsselt.

Wie wirkt sich die *Verschlüsselung im Ruhezustand* auf die Daten aus? Durch Verwendung verwalteter Datenträger werden Ihre Daten verschlüsselt, wenn sie auf dem zugrunde liegenden Azure Storage gespeichert werden. Die Daten auf den temporären Datenträgern oder Daten im Speicher der VM werden nicht verschlüsselt. Erst wenn die Daten des Betriebssystems oder auf dem Datenträger auf dem zugrunde liegenden physischen Datenträger im *Ruhezustand* sind, erfolgt ihre Verschlüsselung. Abbildung 14.3 zeigt, wie die Daten verschlüsselt werden, wenn sie auf einem verwalteten Datenträger gespeichert werden.

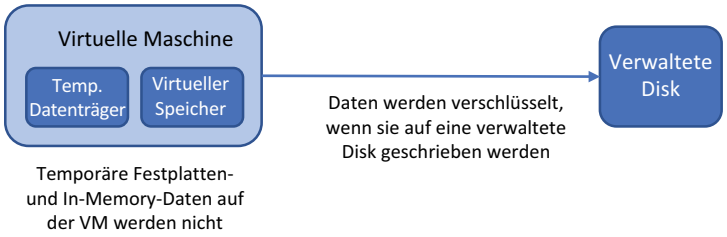


Abbildung 14.3 Daten, die auf einem verwalteten Datenträger gespeichert werden, sind verschlüsselt. In-Memory-Daten auf der VM oder Daten auf temporären Datenträgern, die sich lokal zur VM befinden, werden nicht verschlüsselt. Es sei denn, die gesamte VM ist zur Verschlüsselung aktiviert. Wir gehen in Abschnitt 14.4.2 auf diese Möglichkeit ein. Die automatische Verschlüsselung von Daten, die auf verwalteten Datenträgern gespeichert werden, verursacht keinen zusätzlichen Verwaltungsaufwand für die VM. Die Azure-Plattform führt die Verschlüsselung auf dem zugrunde liegenden Storage durch. Die VM muss keine Verschlüsselungs-/Entschlüsselungsprozesse bearbeiten.

Diese Verschlüsselung ruhender Daten für verwaltete Datenträger führt dazu, dass es zu keinen Auswirkungen auf die Leistung der VMs kommt. Eine zusätzliche Verarbeitung, die die VM durchführen muss, um die Daten zu verschlüsseln und zu entschlüsseln, entfällt. Auf diese Weise kann die gesamte verfügbare CPU-Leistung zum Ausführen von Anwendungen genutzt werden. In typischen VM-Verschlüsselungsszenarien nutzt die VM eine gewisse Rechenleistung, um die Datenverschlüsselung zu verarbeiten und zu verwalten. Der Kompromiss zur automatischen Verschlüsselung verwalteter Datenträger besteht darin, dass nur das Betriebssystem und die Datenträger geschützt sind. Möglicherweise sind andere In-Memory- oder temporäre Datenträgerdaten auf der VM zu einem Angriffsziel werden.

Microsoft verwaltet die digitalen Schlüssel innerhalb der Azure-Plattform durch die automatische Verschlüsselung verwalteter Datenträger. Dadurch entsteht ein weiterer Kompromiss, da Sie Ihre Daten automatisch verschlüsseln können, ohne Schlüssel erstellen, verwalten, rotieren oder widerrufen zu müssen. Sie müssen jedoch Microsoft vertrauen, wenn es darum geht, diese Schlüssel zu schützen.

14.3 Storage Service Encryption

Automatische Datenträgerverschlüsselung ist eine großartige Möglichkeit, aber wie sieht es aus, wenn Sie Azure Storage als Blob oder File Storage verwenden? Mit Azure Storage Service Encryption (SSE) können Sie Daten auf der Ebene des Storage-Kontos verschlüsseln. Die Daten werden verschlüsselt, während sie auf dem Konto gespeichert werden. Microsoft kümmert sich auch hier um die Schlüssel, sodass der Verwaltungs- oder Konfigurationsaufwand entfällt. Die Azure-Plattform abstrahiert für Sie die Schlüsselgenerierung und -verwaltung. Wenn Sie es wünschen, können Sie Ihre eigenen Schlüssel mit einem geringen zusätzlichen Verwaltungsaufwand erstellen und verwenden. Wie bei automatischen verwalteten Verschlüsselung von Datenträgern im Ruhezustand wird die Azure Storage Encryption automatisch aktiviert, wenn Sie ein Konto erstellen.

Sowohl bei der automatischen Verschlüsselung verwalteter Datenträger als auch bei der SSE besteht das Ziel darin, die Verschlüsselung Ihrer Daten so einfach wie möglich zu gestalten und Ihnen mehr Zeit für die Entwicklung und den Betrieb Ihrer Anwendungen zu geben. Abbildung 14.4 zeigt, wie SSE Ihre Daten schützt und auch eine sichere Kommunikation bei der Übertragung Ihrer Daten erzwingen kann.

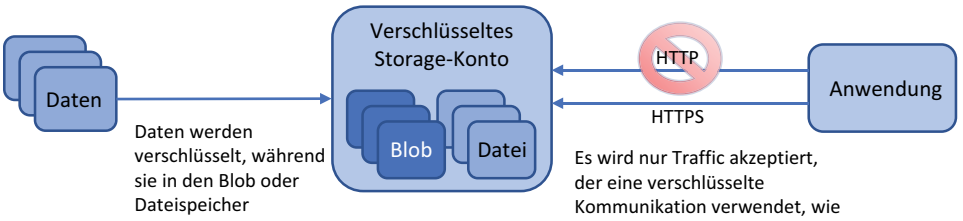


Abbildung 14.4 Wenn Sie SSE aktivieren, werden Azure-Blobs und -Dateien verschlüsselt, während die Daten auf dem Datenträger gespeichert werden. Azure Tables und Azure Queues sind nicht verschlüsselt. Zur zusätzlichen Datensicherheit können Sie die gesamte Kommunikation mit einem Storage-Konto erzwingen, um sichere Kommunikationsprotokolle wie HTTPS zu verwenden. Dadurch werden die übertragenen Daten bis zum Zeitpunkt der Verschlüsselung auf dem Datenträger geschützt.

Erzwingen von sicheren Übertragungen beim Storage Traffic

Zusammen mit der SSE-Aktivierung können Sie alle Speicheranfragen und Übertragungen erzwingen, um eine sichere Kommunikationsmethode zu verwenden. Diese Einstellung erzwingt, dass alle REST-API-Aufrufe HTTPS verwenden und alle Azure-Dateiverbindungen ohne Verschlüsselungsmöglichkeiten, wie beispielsweise ältere Versionen des SMB-Protokolls, gelöscht werden.

(Fortsetzung)

Azure SDKs, wie die Python-Beispiele, die wir in Kapitel 4 untersucht haben, können verschlüsselte Verbindungen verwenden. Die Referenzdokumente für jedes sprachspezifische SDK enthalten Hinweise zur Implementierung einer sicheren Kommunikation.

Die Nutzung sicherer Kommunikationsmöglichkeiten sollte von Anfang an in Anwendungen integriert werden. Bei der Bereitstellung einer sicheren Kommunikation mit einer bestehenden Anwendung kann es zu Problemen kommen, wenn einige Komponenten ursprünglich nicht ordnungsgemäß konfiguriert wurden. Testen Sie zumindest sichere Kommunikationsmöglichkeiten bei einer bestehenden Anwendung zunächst in einer Entwicklungsumgebung.

Jetzt testen

Führen Sie die folgenden Schritte durch, um ein Speicherkonto zu erstellen und sowohl Verschlüsselung als auch sichere Kommunikation zu aktivieren:

1. Öffnen Sie das Azure Portal, und wählen Sie das Cloud Shell-Symbol aus dem oberen Menü.
2. Erstellen Sie eine Ressourcengruppe und geben Sie einen Namen, wie `azuremolchapter14`, und einen Standort, wie `eastus`, ein:

   ```
   az group create --name azuremolchapter14 --location eastus
   ```

3. Erstellen Sie ein Storage-Konto mit `az storage account create`. Geben Sie einen eindeutigen Namen wie z. B. `azuremolstorage` und die Ressourcengruppe ein, die Sie im vorherigen Schritt erstellt haben. Geben Sie einen Storage-Konto-Typ wie `Standard_LRS` für lokal redundante Speicher ein. Um sichere Kommunikation zu erzwingen, stellen Sie `--https-only` ein.

   ```
   az storage account create \
     --name azuremolstorage \
     --resource-group azuremolchapter14 \
     --sku standard_lrs \
     --https-only true
   ```

4. Überprüfen Sie, ob das Storage-Konto verschlüsselt und für eine sichere Kommunikation aktiviert ist, indem Sie `enableHttpsTrafficOnly` und den Verschlüsselungsparametern abfragen:

   ```
   az storage account show \
     --name azuremolstorage \
     --resource-group azuremolchapter14 \
     --query [enableHttpsTrafficOnly,encryption]
   ```

   Die Ausgabe kann wie folgt aussehen:

   ```
   [
     true,
     {
       "keySource": "Microsoft.Storage",
   ```

```
    "keyVaultProperties": null,
    "services": {
      "blob": {
        "enabled": true,
        "lastEnabledTime": "2019-09-27T03:33:17.441971+00:00"
      },
      "file": {
        "enabled": true,
        "lastEnabledTime": "2019-09-27T03:33:17.441971+00:00"
      },
      "queue": null,
      "table": null
    }
  }
]
```

14.4 VM-Verschlüsselung

Die automatische Verschlüsselung von verwalteten Azure-Datenträgern gewährleistet ein hohes Maß an VM-Sicherheit. Für einen umfassenden Ansatz zur VM-Datensicherheit können Sie die VM selbst verschlüsseln. Dieser Prozess umfasst mehr als die Verschlüsselung der zugrunde liegenden virtuellen Datenträger. Der Datenträger des Betriebssystems und alle angeschlossenen Datenträger sowie der temporäre Datenträger sind verschlüsselt. Der VM-Arbeitsspeicher ist ebenfalls verschlüsselt, um die Angriffsfläche weiter zu minimieren. Sie verwenden digitale Schlüssel, um VMs zu verschlüsseln.

Ein Vorteil der Verschlüsselung der gesamten VM besteht darin, dass Sie die Schlüssel verwalten. Diese Schlüssel werden sicher in Azure Key Vault gespeichert, und Sie können zwischen der Verwendung von Software- oder Hardware-generierten Schlüsseln wählen. Sie steuern diese Schlüssel, können den Zugriff darauf definieren und rollenbasierte Zugriffskontrollen und Überwachung zur Rückverfolgung der Nutzung einsetzen. Sie können die Schlüssel auch nach einem definierten Zeitplan rotieren und Ihr Passworts alle 60 oder 90 Tage ändern. Diese zusätzlichen Kontrollen und Verwaltungsaufgaben für Schlüssel erhöhen den Verwaltungsaufwand, bieten aber maximale Flexibilität bei der Sicherung Ihrer Daten und können für bestimmte regulatorische Zwecke erforderlich sein. Sehen wir uns ein wenig mehr von Azure Key Vault an.

14.4.1 Speicherung der Schlüssel in Azure Key Vault

Wir haben uns im Kapitel 15 mit Azure Key Vault beschäftigt, aber zunächst wollte ich Ihnen das Potenzial der Datenverschlüsselung und VM-Verschlüsselung demonstrieren. Fangen wir mit einer kurzen Übersicht an. Bei Azure Key Vault handelt es sich um einen digitaler Tresor, mit dem Sie Schlüssel, SSL-Zertifikate und geheime Daten wie Passwörter sicher speichern können. Aus Redundanzgründen werden wichtige Datenspeicher in alle Azure-Regionen repliziert. Diese Replikation schützt Ihre Schlüssel und geheime Daten und stellt sicher, dass sie immer einsatzbereit sind.

Nur Sie haben Zugriff auf Ihre Schlüsseltresore. Sie generieren und speichern Objekte in Schlüsseltresoren und definieren dann, wer Zugriff auf diese Tresore hat. Microsoft verwaltet den zugrunde liegenden Key Vault-Dienst, verfügt aber über keinen Zugriff auf den Inhalt des Tresors. Diese Sicherheitsschranke bedeutet, dass wenn Sie Ihre Daten auf Azure verschlüsseln, Sie der Einzige sind, der sie entschlüsseln und einsehen kann.

Jetzt testen

Führen Sie die folgenden Schritte aus, um einen Schlüsseltresor und einen Schlüssel zu erstellen:

1. Öffnen Sie das Azure Portal, und wählen Sie das Cloud Shell-Symbol aus dem oberen Menü.
2. Geben Sie einen Schlüsseltresor mit dem Befehl `az keyvault create` an, geben Sie die Ressourcengruppe an, die Sie in der vorherigen Übung erstellt haben, z. B. azuremolchapter14, und geben Sie dann einen eindeutigen Namen für Ihren Schlüsseltresor, z. B. `azuremolkeyvault` an:

```
az keyvault create \
  --resource-group azuremolchapter14 \
  --name azuremolkeyvault \
  --enabled-for-disk-encryption
```

Lassen Sie uns eine Pause einlegen und darüber nachdenken, warum Sie einen Parameter für `--enabled-for-disk-encryption` einfügen. Wenn Sie eine VM verschlüsseln, muss die Azure-Plattform die VM starten und entschlüsseln können, damit sie ausgeführt werden kann. Die Azure-Plattform verfügt über keine Berechtigungen für den Zugriff auf diese Daten, und Microsoft besitzt keinen Zugriff auf die Anzeige und Verwendung dieser Schlüssel für andere Zwecke als für das Starten einer VM. Wenn Sie einen Schlüsseltresor zur Datenträgerverschlüsselung aktivieren, erteilen Sie Azure die Berechtigung, auf den Schlüsselspeicher zuzugreifen und den einer VM zugeordneten Schlüssel zu verwenden.

Auch hier hat Microsoft keinen Zugriff auf diese Schlüssel oder Ihre Daten, sondern nur die Möglichkeit, Ihre verschlüsselte VM zu starten. Eine verschlüsselte VM, die nicht booten kann, bietet nicht viele Möglichkeiten, um damit zu arbeiten. Abbildung 14.5 zeigt, wie die Azure-Plattform den Schlüssel verwendet, um eine verschlüsselte VM zu starten.

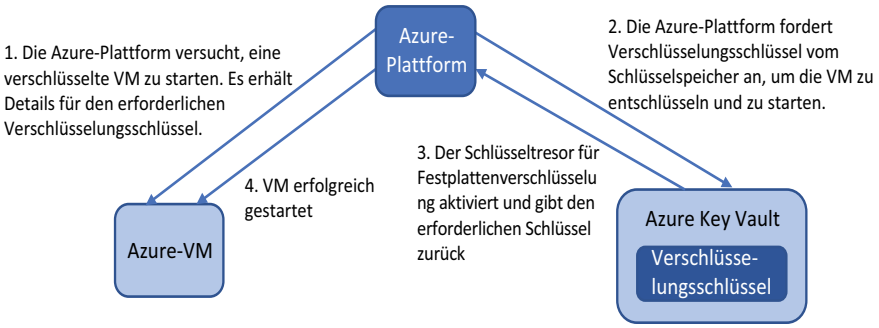


Abbildung 14.5 Wenn ein Schlüsseltresor für die Datenträgerverschlüsselung aktiviert ist, erteilt er der Azure-Plattform die Berechtigung, den Schlüssel anzufordern und zu verwenden, um eine verschlüsselte VM erfolgreich zu starten.

Schlüssel können innerhalb der Software erstellt und gespeichert werden, oder sie können in Hardware-Sicherheits-Modulen (HSM) für zusätzliche Sicherheit gespeichert werden. Für viele Zwecke funktionieren Softwareschlüssel hervorragend, obwohl Sie möglicherweise an Sicherheitsauflagen gebunden sind, die den Einsatz von HSM erfordern. Wir werden dieses Thema im Kapitel 15 näher erläutern.

3 Um einen Schlüssel zu erstellen, geben Sie den Tresor an, den Sie in Schritt 2 erstellt haben, z. B. azuremolkeyvault, und geben Sie dann einen Schlüsselnamen an, z. B. `azuremolencryptionkey`:

```
az keyvault key create \
  --vault-name azuremolkeyvault \
  --name azuremolencryptionkey \
  --protection software
```

14.4.2 Verschlüsseln einer Azure VM

Der Verschlüsselungsschlüssel, den Sie in Abschnitt 14.4.1 erstellt haben, kann verwendet werden, um bei Bedarf viele VMs zu verschlüsseln. Dadurch wird der Aufwand für die Schlüsselverwaltung minimiert, und Sie können die Anzahl der VM-Instanzen automatisch skalieren, wenn Sie Skalen-Sets für Virtual Machine verwenden ohne jedes Mal Schlüssel generieren zu müssen. Die Alternative besteht darin, dass jede VM über einen eigenen Verschlüsselungsschlüssel verfügt, der die Komplexität erhöht, aber eine Sicherheitsebene für Ihre VMs bietet. Wenn Sie den gleichen Verschlüsselungsschlüssel für Back-End-Anwendungs-VMs und dann Datenbank-VMs verwenden, könnte ein theoretischer Angreifer mit diesem einen Schlüssel Zugriff auf die Daten beider Gruppen von VMs erhalten. Wenn verschiedene Schlüssel verwendet werden, ist die Anzahl der potenziell kompromittierten VMs niedriger. In der Übung am Ende des Kapitels werden Sie eine einzelne VM verschlüsseln, obwohl der gleiche Prozess mit einer Skalierungsgruppe, die mehrere VMs aufweist, nur den einen Schlüssel verwendet. Besonders wenn Sie mit größeren, automatisch skalierenden Anwendungen arbeiten, sollten Sie unbedingt Sicherheitsfunktionen entwickeln und einbauen.

Es wird eine Azure VM-Erweiterung installiert, wenn Sie eine VM verschlüsseln. Die Erweiterung steuert die Verschlüsselung des Betriebssystemdatenträgers, des temporären Datenträgers, aller angeschlossenen Datenträger und der In-Memory-Daten, wie dies in Abbildung 14.6 dargestellt ist. Für Windows-VMs wird der BitLocker-Verschlüsselungsmechanismus verwendet. Bei Linux-VMs wird dm-crypt zur Verarbeitung der Verschlüsselung verwendet. Die VM-Erweiterung kann dann über den Status der Verschlüsselung Bericht erstatten und die VM wie gewünscht entschlüsseln.

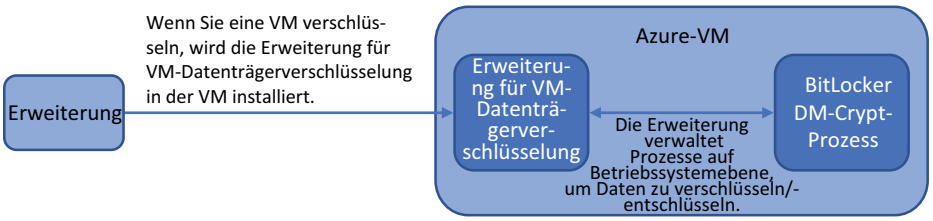


Abbildung 14.6 Es wird eine Azure-Datenträgererweiterung installiert, wenn Sie eine VM verschlüsseln. Diese Erweiterung verwaltet die Verwendung von BitLocker auf Windows-VMs oder dm-crypt auf Linux-VMs, um die Datenverschlüsselung auf Ihrer VM durchzuführen. Die Erweiterung wird auch verwendet, wenn Sie den Verschlüsselungsstatus für eine VM abfragen.

Da die Erweiterung der VM-Datenträgerverschlüsselung BitLocker oder dm-crypt als Grundlage nutzt, bestehen einige Einschränkungen bei der Verwendung der VM-Verschlüsselung. Die meisten Azure Marketplace-Images unterstützen Datenträgerverschlüsselung, obwohl einige Einschränkungen bei VM-Größen bestehen, die die Verschlüsselung oder Verschlüsselung von vernetzten Netzwerkdateifreigaben wie Azure-Dateien unterstützen. Umfassende Informationen zu unterstützten Einschränkungen und Überlegungen zur VM-Verschlüsselung finden Sie in den neuesten Azure-Dokumenten unter http://mng.bz/yyvd.

Dieses Kapitel bietet eine kurze Einführung in die Funktionen für Datensicherheit und Verschlüsselung auf Azure. Die automatische Verschlüsselung für verwaltete Datenträger und SSE erfordert nicht viel Konfiguration. Es gibt daher keine wirklichen Hürden, die Sie an ihrer Verwendung hindern.

14.5 Übung: Verschlüsseln einer VM

Lassen Sie uns all dies in Aktion erleben, indem Sie eine VM mit einem Schlüssel verschlüsseln, den Sie in Ihrem Schlüsseltresor gespeichert haben:

1. Erstellen Sie eine VM. Die meisten Linux-Images auf dem Azure Marketplace unterstützen Verschlüsselung, ebenso wie die Windows Server-Images von Windows Server 2008 R2 und höher. Um diesen Prozess schnell und einfach zu gestalten, erstellen Sie eine Ubuntu LTS VM, wie sie es anhand der Anleitung aus diesem Buch bereits getan haben. Da die VM ausreichend Arbeitsspeicher benötigt, um den Datenträgerverschlüsselungsvorgang durchzuführen, geben Sie eine Größe von Standard_D2s_v3 an:

```
az vm create \
  --resource-group azuremolchapter14 \
  --name molvm \
  --image ubuntults \
  --size Standard_D2s_v3 \
  --admin-username azuremol \
  --generate-ssh-keys
```

2. Aktivieren Sie die Verschlüsselung auf der VM, und geben Sie den Namen des Azure Key Vault sowie den digitalen Schlüssel an, den Sie in einer früheren Übung erstellt haben:

```
az vm encryption enable \
  --resource-group azuremolchapter14 \
  --name molvm \
  --disk-encryption-keyvault azuremolkeyvault \
  --key-encryption-key azuremolencryptionkey
```

   Es dauert ein paar Minuten, um die Erweiterung zur Azure VM-Datenträgerverschlüsselung zu installieren und den Verschlüsselungsprozess für die VM zu starten.

3 Nachdem die Verschlüsselung eingeleitet wurde, überwachen Sie den Fortschritt, und starten Sie die VM neu, um den Verschlüsselungsprozess abzuschließen. Der Status wird wie folgt angezeigt:

```
az vm encryption show \
--resource-group azuremolchapter14 \
--name molvm \
--query 'status'
```

Hier ist ein Beispiel für die Ausgabe einer VM im Verschlüsselungsprozess. Zu Beginn werden die Statusmeldungen angezeigt als

```
[
  {
    "code": "ProvisioningState/succeeded",
    "displayStatus": "Provisioning succeeded",
    "level": "Info",
    "message": "OS disk encryption started",
    "time": null
  }
]
```

Es kann einige Zeit in Anspruch nehmen, bis die Datenträgerverschlüsselung abgeschlossen ist. Es ist eine gute praktische Übung, auf die wir in einer Stunde oder so zurückkommen können, es sei denn, Sie möchten eine lange Mittagspause einlegen! Ich bin *zwar* nicht Ihr Chef, doch es ist ziemlich langweilig, dieselbe Meldung zum Verschlüsselungsstatus so lange anzustarren.

4 Wenn der Verschlüsselungsstatus als `Verschlüsselung für alle Volumes` erfolgreich gemeldet wird, starten Sie die VM neu:

```
az vm restart --resource-group azuremolchapter14 --name molvm
```

Sie können dann den Status der VM-Verschlüsselung noch einmal mit `az vm encryption show` überprüfen, um zu bestätigen, dass die VM-Verschlüsselung verschlüsselt gemeldet wird.

Denken Sie an Ihre Hausreinigungsarbeiten

Diese letzten beiden praktischen Übungen am Ende des Kapitels haben nicht viel Zeit in Anspruch genommen, aber es kann eine Weile gedauert haben, um die Prozesse abzuschließen. Vergessen Sie nicht, zurückzugehen und Ressourcen zu löschen, wenn Sie damit fertig sind.

Denken Sie in Kapitel 13 daran, dass Sie den Schutz von Azure Backup oder Site Recovery deaktivieren müssen, bevor Sie den Tresor und die Ressourcengruppe von Recovery Services löschen können (nachdem Sie die 14 Tage gewartet haben, bis die kostenlosen Soft-Delete-Wiederherstellungspunkte abgelaufen sind). Sie müssen unbedingt zurückgehen und diese Laborressourcen entfernen, bevor sie zu viel von Ihrem Azure-Guthaben verbrauchen.

15 Sicherung von Daten mit Azure Key Vault

Fast jede Woche gibt es Nachrichten über einen Cybersecurity-Vorfall bei einem großen Unternehmen. Die Angreifer automatisieren ihre eigenen Aktionen genau so, wie Sie die verschiedene Formen der Automatisierung genutzt haben, um Ihre Anwendungen und Daten zu erweitern oder zu replizieren. Es ist unwahrscheinlich, dass eine Einzelperson versuchen wird, die Sicherheit Ihrer Systeme zu manipulieren. Genau deswegen ist es so schwierig, Systeme rund um die Uhr 365 Tage im Jahr (na gut, oder 366 Tage!) zu beschützen.

In Kapitel 14 ging es darum, wie Sie Ihre Daten und VMs verschlüsseln können. Dies ist ein großartiger erster Schritt. Außerdem haben wir uns kurz mit der Erstellung und Verwendung von Verschlüsselungscodes beschäftigt, die auf dem Azure Key Vault-Dienst gespeichert werden. Sichere Daten, wie Schlüssel, Geheimnisse und Zertifikate, werden am besten in einem digitalen Tresor wie einem Schlüsseltresor gespeichert, der die Verwendung Ihrer kritischen Anmeldeinformationen und Daten zentral verwalten, ausgeben und überprüfen kann. Da Ihre Anwendungen und Dienste Zugriff auf verschiedene Ressourcen haben müssen, können sie diese Schlüssel, Geheimnisse und Anmeldeinformationen automatisch anfordern, abrufen und verwenden. In diesem Kapitel erfahren Sie mehr darüber, warum und wie Sie einen sicheren Schlüsselspeicher erstellen, den Zugriff darauf kontrollieren und dann Geheimnisse und Zertifikate speichern und abrufen können.

15.1 Die Sicherung von Daten in der Cloud

Anwendungen werden komplexer und das Risiko von Cyberangriffen steigt. Die Sicherheit wird zu einem entscheidenden Teil der Art und Weise, wie Sie Ihre Dienste entwerfen und betreiben. Gerade wenn Sie mehr internetfähigen Anwendungen On-Premises oder in der Cloud ausführen, muss die Minimierung des Risikos eines unbefugten Datenzugriffs einer der Schwerpunkte Ihrer Arbeit sein. Es ist sinnlos, die größte Pizzeria der Welt zu haben, wenn Kund*innen Ihnen ihre Zahlungsdaten oder persönlichen Daten nicht anvertrauen.

Oftmals wird die Sicherheit für Anwendungen und Dienste anhand digitaler Schlüssel, Geheimnisse und Zertifikate, wie in Abbildung 15.1, gewährleistet. Anstatt

einen Benutzungsnamen und ein Passwort zu verwenden, die immer wieder manuell eingegeben werden müssen, – oder, schlimmer noch, in einer unverschlüsselten Konfigurationsdatei aufbewahrt sind – verwenden Sie einen digitalen Tresor, um Ihre sicheren Anmeldeinformationen und Daten zu speichern. Wenn eine Anwendung oder ein Dienst Zugriff benötigt, fordert sie den spezifischen Schlüssel oder das Geheimnis an, und es wird auch ein Überwachungstrail erstellt, der jeden möglichen Sicherheitsmissbrauch oder Verletzung der Sicherheit verfolgt.

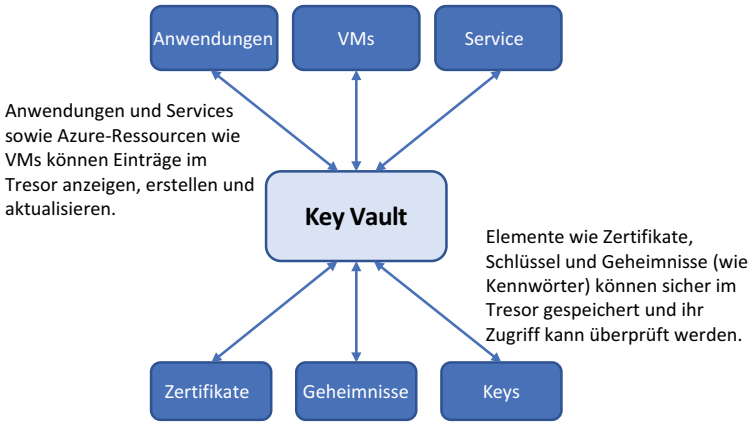


Abbildung 15.1 Azure Key Vault bietet eine sichere Möglichkeit, digitale Informationen wie Zertifikate, Schlüssel und Geheimnisse zu speichern. Diese Anwendungen, Dienste oder Azure-Ressourcen wie VMs können dann auf diese sicheren Elemente direkt zugreifen. Mit minimaler menschlicher Interaktion können Sie sichere Anmeldeinformationen und Zertifikate zentral über Ihre Anwendungsumgebungen verteilen.

Bei richtiger Planung und Implementierung sind diese digitalen Tresore nahezu vollständig automatisiert und sicher. Dienste können ein neues digitales Zertifikat anfordern, das dann sicher im Tresor gespeichert wird. Damit können sie sich dann anderen Anwendungskomponenten gegenüber autorisieren. Server können Software konfigurieren, indem sie Geheimnisse wie Kennwörter aus dem digitalen Tresor abrufen und dann Anwendungskomponenten installieren, ohne dass die Anmeldeinformationen in einer textbasierten Konfigurationsdatei gespeichert werden. Ein*e Anwendungsadministrator*in kann alle Geheimnisse, Schlüssel und Zertifikate eines Dienstes zentral verwalten und bei Bedarf regelmäßig aktualisieren.

Azure Key Vault bietet all diese digitalen Sicherheitsfunktionen und ermöglicht es Ihnen, genau zu kontrollieren, welche Benutzer*innen und Ressourcen auf die sicheren Daten zugreifen können. Wichtige Datenspeicher können sicher repliziert werden, um Redundanzen und eine verbesserte Anwendungsleistung zu gewährleisten und mit gängigen Azure-Ressourcen wie VMs, Webanwendungen und Azure Storage-Konten zu integrieren.

15.1.1 Softwaretresore und Hardwaresicherheitsmodule

Bevor wir uns ein praktisches Beispiel ansehen, wie man einen Schlüsseltresor erstellt und verwendet, ist es wichtig zu verstehen, wie Ihre sicheren Informationen in einem Tresor gespeichert werden. Wie in Abbildung 15.2 dargestellt, werden alle Schlüssel, Geheimnisse und Zertifikate in einem Schlüsselspeicher in einem Hardwaresicherheitsmodul (HSM) gespeichert. Diese Geräte sind nicht nur für Azure erhältlich – es gibt Hardwaregeräte, die branchenweit eingesetzt werden und die ein hohes Maß an Sicherheit für alle auf ihnen gespeicherten Daten bieten.

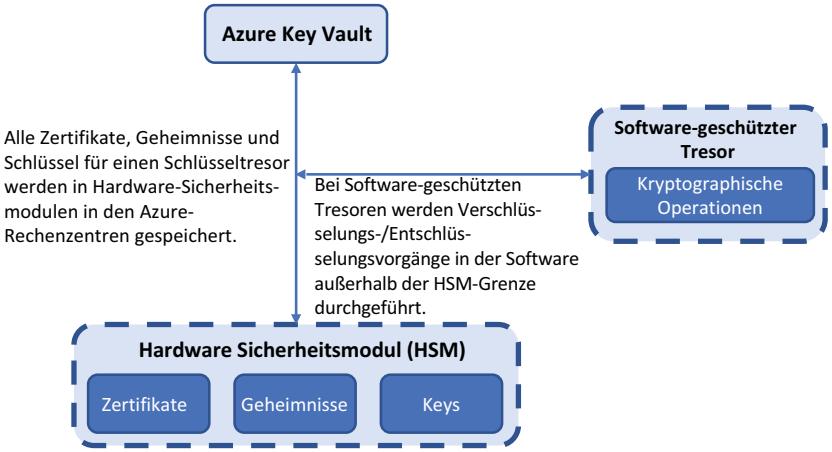


Abbildung 15.2 Azure Key Vault ist eine logische Ressource auf Azure, doch alle Zertifikate, Geheimnisse und Schlüssel werden in einem HSM gespeichert. Für Entwicklungs- oder Testszenarien kann ein softwaregeschützter Tresor verwendet werden, der dann alle kryptographischen Operationen – wie das Ver- und Entschlüsseln von Daten – in seiner Software durchführt und nicht über die Hardware im HSM. Für die Produktion sollten Sie einen HSM-geschützten Tresor verwenden, über den die gesamte Verarbeitung auf der Hardware erfolgt.

Es gibt derzeit zwei verschiedene Arten von Schlüsseltresoren: softwaregeschützt und HSM-geschützt. Diese Unterscheidung kann verwirrend sein, weshalb ich sie kurz erklären möchte, bevor wir anfangen:

- Ein *softwaregeschützter* Tresor speichert Schlüssel, Geheimnisse und Zertifikate in einem HSM. Jedoch werden alle kryptographischen Operationen, die zum Verschlüsseln oder Entschlüsseln seines Inhalts erforderlich sind, von der Azure-Plattform anhand von Software durchgeführt. Softwaregeschützte Tresore eignen sich hervorragend für Entwicklungs- und Testszenarien. Obwohl Sie sich natürlich auch entscheiden können, dass Produktionsworkloads eine etwas sicherere Methode benötigen, um die kryptografischen Vorgänge durchzuführen.
- Ein *HSM-geschützter Tresor* speichert Schlüssel, Geheimnisse und Zertifikate in einem HSM, doch kryptografische Vorgänge, die zum Ver- oder Entschlüsseln seines Inhalts erforderlich sind, werden direkt im HSM ausgeführt. Sie können auch Ihre eigenen sicheren Schlüssel in einem On-Premises HSM generieren und diese dann in Azure importieren. Es gibt einige zusätzliche Tools und Prozesse, die befolgt werden müssen, doch auf diese Weise stellen Sie sicher, dass Sie die Schlüssel vollständig kontrollieren und dass sie zu keinem Zeitpunkt das HSM verlassen.

  Um die Sicherheit und Integrität Ihrer Daten zu maximieren, sind hardwaregeschützte Tresore der bevorzugte Ansatz für Workloads in der Produktion.

Unabhängig davon, welche Art von Tresor Sie verwenden, ist es wichtig, daran zu denken, dass alle Ihre Daten sicher auf einem Computer gespeichert sind. Der Federal Information Processing Standard (FIPS) 140-2 Level 2 validiert (mindestens) das HSM, und dass Microsoft keinen Zugriff auf Ihre Schlüssel hat. Für HSM-geschützte Tresore fallen zusätzliche Kosten an. Wie bei allen Azure- und Cloud-Computing-Lösungen sollten Sie also die Kosten gegen das Risiko abwägen, dass Ihre Daten gefährdet werden.

15.1.2 Erstellen eines Schlüsseltresors und eines Geheimnisses

Ein digitaler Tresor klingt zwar großartig, aber Sie sind sich vielleicht etwas unsicher, wie Sie die Vorteile von Azure Key Vault richtig nutzen können. Lassen Sie uns ein Beispiel für einen einfachen Server erstellen, der eine Datenbank wie einen MySQL Server betreibt, wie in Abbildung 15.3 dargestellt.

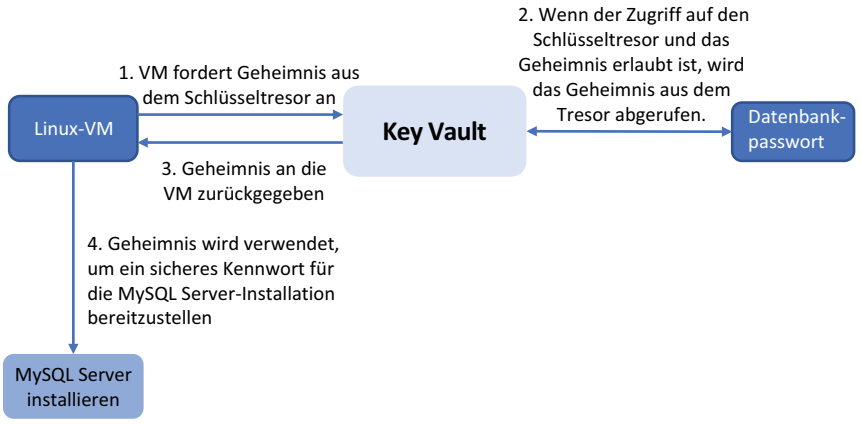


Abbildung 15.3 In den nächsten Übungen erstellen Sie ein Beispiel für ein in einem Schlüsselspeicher gespeichertes Geheimnis, das als Datenbankpasswort für eine MySQL Server-Installation verwendet werden kann. Es wird eine VM erstellt, die über die Berechtigung verfügt, das Geheimnis aus dem Schlüsseltresor anzufordern. Das abgerufene Geheimnis wird dann verwendet, um während des Installationsprozesses der Anwendung automatisch eine sichere Anmeldeinformationen einzugeben.

Eine der ersten Übungen in diesem Buch war, eine VM zu erstellen und dann den LAMP-Webserver-Stack zu installieren. Sie wurden wahrscheinlich nach einem MySQL Server-Passwort gefragt, oder es wurde automatisch ein leeres Passwort verwendet. Nachdem Sie nun alles über die wichtigsten Tresore wissen, können Sie automatisch ein Passwort aus dem Tresor abrufen und es dynamisch zur Installation und Konfiguration des Servers verwenden.

Jetzt testen

Um einen Schlüsseltresor zu erstellen und ein Geheimnis hinzuzufügen, führen Sie die folgenden Schritte aus:

1. Öffnen Sie das Azure Portal. Starten Sie Cloud Shell, und erstellen Sie eine Ressourcengruppe, z. B. `azuremolchapter15`:

```
az group create --name azuremolchapter15 --location eastus
```

2. Erstellen Sie einen Schlüsseltresor mit einem eindeutigen Namen, wie z. B. `azuremol`, und aktivieren Sie ihn für die Bereitstellung, sodass Sie den Tresor verwenden können, um Schlüssel und Zertifikate in eine VM einzubringen::

```
az keyvault create \
  --resource-group azuremolchapter15 \
  --name azuremol \
```

```
  --enable-soft-delete \
  --enabled-for-deployment
```

Standardmäßig ist Ihrem Azure-Benutzungskonto die volle Berechtigung für den Schlüsseltresor zugewiesen. Für diese Übungen ist dies in Ordnung, obwohl Sie sich später als Sicherheitsmaßnahme überlegen sollten, wer auf Ihren Schlüsseltresor zugreifen darf. Sie können die `--no-self-perms`-Parameter hinzufügen, um die Berechtigungsvergabe an Ihr Konto zu überspringen.

3. Erstellen Sie ein Geheimnis, wie z. B. `databasepassword` und weisen Sie ihm einen Kennwortwert zu, z. B. `SecureP@ssw0rd`. (Ja, sehr sicher, nicht wahr?) Dieses Geheimnis kann als Anmeldeinformationen für einen Datenbankserver verwendet werden, den Sie in den folgenden Übungen einsetzen:

```
az keyvault secret set \
  --name databasepassword \
  --vault-name azuremol \
  --description "Database password" \
  --value "SecureP@ssw0rd"
```

4. Sie haben volle Berechtigungen für den Schlüsseltresor, sodass Sie den Inhalt Ihres Geheimnisses einsehen können:

```
az keyvault secret show \
--name databasepassword \
--vault-name azuremol
```

Aus der Verwaltungsansicht können Sie auch allgemeine Aktionen wie das Sichern und Wiederherstellen, Herunterladen, Aktualisieren und Löschen von Elementen, die in einem Schlüsseltresor gespeichert sind, durchführen. Eine weitere Eigenschaft, die Sie beim Erstellen des Schlüsselspeichers festgelegt haben, ist die Option `enable-soft-delete`. Wenn Ihre Anwendungen und Dienste die Geheimnisse nicht abrufen können, die sie aus dem Schlüsseltresor benötigen, könnte auf Sie ein ziemlich großer Anwendungsausfall zukommen! Ein Schlüsseltresor kann Metadaten für Geheimnisse bis zu 90 Tage lang speichern, nachdem sie wirklich gelöscht wurden, sodass Sie Daten wiederherstellen können, die falsch oder böswillig gelöscht wurden.

5. Löschen Sie den Schlüssel, den Sie gerade erstellt haben, um einen Fehler oder die Aktionen einer Person, die mit böser Absicht handelt, zu simulieren:

```
az keyvault secret delete \
  --name databasepassword \
  --vault-name azuremol
```

6. Stellen Sie das Geheimnis wieder her, damit Sie das Datenbankpasswort mit Ihrer Anwendung und Ihren Diensten weiterhin verwenden können:

```
az keyvault secret recover \
  --name databasepassword \
  --vault-name azuremol
```

Wenn Sie ein Geheimnis wirklich entfernen möchten, gibt es auch die Möglichkeit, ein gelöschtes Geheimnis zu beseitigen. Diese Option entfernt das Geheimnis dauerhaft, ohne auf die standardmäßigen 90 Tage der Wiederherstellungszeit zu warten.

Zögern Sie nicht, `az keyvault secret show` erneut zu nutzen, um die Informationen über Ihr Geheimnis einzusehen und zu bestätigen, dass das von Ihnen gespeicherte Kennwort vorhanden ist. Lassen Sie uns nun weitermachen, um zu sehen, wie eine VM auf einen Schlüsselspeicher zugreifen kann und das Geheimnis zur Installation des MySQL Servers nutzen kann.

15.2 Verwaltete Identitäten für Azure-Ressourcen

Die Möglichkeit, Azure Key Vault zu benutzen, um Geheimnisse oder Schlüssel zu speichern, ist großartig, aber wie greifen Sie wieder auf diese Geheimnisse zu? Die Azure CLI oder Azure PowerShell kann auf die in einem Schlüsselspeicher gespeicherten Informationen zugreifen. Allerdings ist es oftmals bequemer, Ihren VMs oder Anwendungen zu ermöglichen, Geheimnisse oder Schlüssel direkt abzurufen, wenn sie diese benötigen. Eine Variante, dies zu tun, ist mit verwalteten Identitäten für Azure-Ressourcen, wie in Abbildung 15.4 dargestellt.

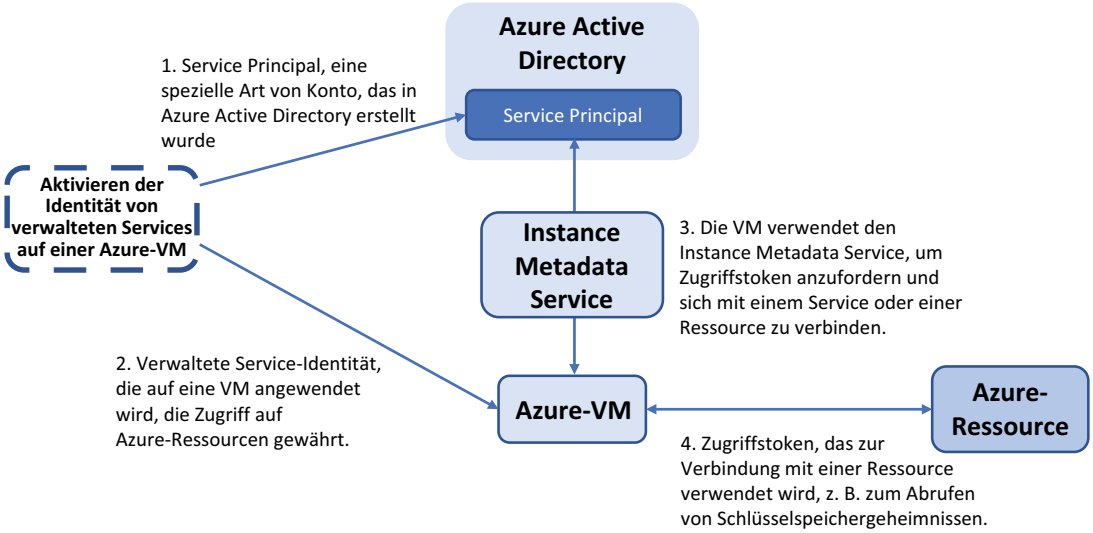


Abbildung 15.4 Wenn Sie eine verwaltete Identität für eine VM erstellen, wird im Azure Active Directory ein Service Principal erstellt. Dieses Dienstprinzipals ist eine spezielle Art von Konto, das für Ressourcen zur Authentifizierung verwendet werden kann. Diese VM verwendet dann den Endpunkt des Instance Metadata Services, um Anfragen für den Ressourcenzugriff zu stellen. Der Endpunkt verbindet sich mit dem Azure AD, um Zugriffstoken anzufordern, wenn die VM Daten von anderen Diensten anfordern muss. Wenn ein Zugriffstoken zurückgegeben wird, kann dieser verwendet werden, um den Zugriff auf Azure-Ressourcen, wie beispielsweise einen Schlüsseltresor, anzufordern.

Mit einer *verwalteten Identität* können Sie eine spezielle Art von Konto erstellen, die von einer Azure-Ressource, z. B. einer VM, verwendet werden kann. Wenn Sie einen Verzeichnisdienst wie Active Directory verwendet haben, wird häufig ein Computerkonto verwendet, um verschiedene Netzwerkressourcen zu identifizieren und einem Computer Zugang zu gewähren. Sie erstellen und verwenden keine regulären Benutzungskonten für diese Art der Authentifizierung. Dadurch wird die Sicherheit verbessert: Sie können einem Computer nur restriktive Berechtigungen erteilen, anstatt sich beispielsweise um Benutzungsberechtigungen und den Zugriff auf freigegebene Ordner kümmern zu müssen.

Eine verwaltete Identität ist wie ein Computerkonto, aber sie wird in Azure Active Directory (Azure AD) gespeichert. Die Identity, die wir ein *Service Principal* nennen, ist für jede VM eindeutig und kann verwendet werden, um Berechtigungen für andere Azure-Ressourcen zuzuweisen, wie z. B. ein Azure Storage-Konto oder einen Schlüsseltresor. Die VM hat Zugriffsberechtigungen für diese Ressourcen, sodass Sie Aufgaben planen können (z. B. mit Azure Automation, was wir uns in Kapitel 18 näher ansehen), die keinen Benutzereingriff erfordern oder nach Benutzernamen und Kennwörtern fragen. Die VMs authentifizieren sich selbst, und die Azure-Plattform autorisiert den Zugriff auf die ihnen zugewiesenen Ressourcen.

Sie können zwei Arten von verwalteten Identitäten erstellen:

- *Vom System zugewiesen* – dieser Typ der verwalteten Identität wird direkt auf eine Ressource, wie eine VM, angewendet und nur von dieser Ressource verwendet. Jede Ressource hat ihre eigene Identität, wenn es um die Überwachung oder Fehlerbehebung geht. Wenn die Ressource ausgewählt ist, wird die verwaltete Identität automatisch gelöscht.
- *Vom Benutzenden zugewiesen* – eine separate Azure-Ressource wird für die angegebene verwaltete Identität erstellt und verwaltet. Diese verwaltete Identität kann für andere Ressourcen freigegeben werden, um den Zugriff zu definieren. Wenn alle Ressourcen, die die Identität verwenden, gelöscht wurden, bleibt die verwaltete Identität für die Verwendung verfügbar.

Sehen wir einmal nach, wie Sie eine vom System zugewiesene verwaltete Identität verwenden können, um das `databasepassword`-Geheimnis aus einem Schlüsseltresor anzufordern. Sobald die VM das Geheimnis abrufen kann, wird das Passwort verwendet, um automatisch einen MySQL-Datenbankserver zu installieren. Mit einem Schlüsselspeicher und MSI können Sie ein paar Befehle ausführen, um das Geheimnis aus dem Schlüsselspeicher abzurufen, das MySQL Server-Installationsprogramm auszuführen und automatisch das sichere Passwort bereitzustellen.

Azure Instance Metadata Service

Eine VM, die mit einer verwalteten Identität aktiviert wurde, verwendet einen REST-Endpunkt über den Instance Metadata Service (IMDS), um ein Zugriffstoken von Azure AD anzufordern, mit dem sie dann Daten von Azure Key Vault anfordern kann. Aber was ist der Instance Metadata Service?

Der IMDS ist ein REST-Endpunkt, der nur intern für VMs zugänglich ist. Der Endpunkt ist verfügbar unter der nicht routingfähigen Adresse 169.254.169.169.254. Eine VM kann eine Anforderung an den IMDS-Endpunkt senden, um Informationen über sich selbst abzurufen, wie z. B. die Azure-Region oder den Namen der Ressourcengruppe. Dies ermöglicht es der VM zu verstehen, wie und wo sie auf der Azure-Plattform läuft. Der IMDS-Endpunkt kann von vielen vielen Sprachen erreicht werden, darunter Python, C#, Go, Java und PowerShell.

Bei Wartungsereignissen kann auch der IMDS-Endpunkt abgefragt werden, sodass die VM von einem anstehenden Update- oder Neustartereignis Kenntnis erlangt. Alle erforderlichen Aufgaben vor dem Update oder dem Neustart können dann durchgeführt werden. Da IMDS ein REST-Endpunkt auf einer nicht routingfähigen IP-Adresse ist, gibt es keinen Agenten und keine Erweiterung für die zu installierende VM und insofern keine Bedenken hinsichtlich Netzwerksicherheit oder Routing.

Für Zwecke der verwalteten Identität wird der IMDS-Endpunkt verwendet, um die Anfrage nach einem Zugriffstoken an Azure AD weiterzuleiten. Dieser Ansatz bietet VMs eine sichere Möglichkeit, den Zugriff anzufordern, ohne direkt mit dem Azure AD kommunizieren zu müssen.

Jetzt testen

Um eine VM mit einem MSI zu erstellen, führen Sie die folgenden Schritte aus:

1. Erstellen Sie eine Ubuntu-VM. Vergeben Sie dann Ihre Ressourcengruppe, z. B. `azuremolchapter15`, und einen Namen für die VM, wie `molvm`. Ein Benutzungskonto namens azuremol wird erstellt, und die SSH-Schlüssel, die Sie in den vorangegangenen Kapiteln verwendet haben, werden der VM hinzugefügt:

```
az vm create \
  --resource-group azuremolchapter15 \
  --name molvm \
  --image ubuntults \
  --admin-username azuremol \
  --generate-ssh-keys
```

2. Aus Sicherheitsgründen sollten Sie Konten nicht erlauben, auf alle Ressourcen Ihres gesamten Azure-Abonnements zuzugreifen. Insbesondere für verwaltete Identitäten sollten Sie nur die minimale Anzahl von Berechtigungen vergeben, die erforderlich sind.

   Greifen Sie für diese Übung nur auf Ihre Ressourcengruppe zu, wie z. B. auf `azuremolchapter15`. Sie richten den Umfang ein, indem Sie nach der ID der Ressourcengruppe mit `--query id` fragen. Diese ID wird dann einer Variablen namens `scope` zugewiesen:

```
scope=$(az group show --resource-group azuremolchapter15
➥--query id --output tsv)
```

3. Erstellen Sie eine vom System zugewiesene verwaltete Identität für die VM mit der Leserolle, sodass sie nur Ressourcen lesen und keine Änderungen an ihnen vornehmen kann. Legen Sie die Identität für die Ressourcengruppe fest. Die Variable, die Sie im vorherigen Schritt angelegt haben und die die Ressourcengruppen-ID enthält, wird bereitgestellt:

```
az vm identity assign \
  --resource-group azuremolchapter15 \
  --name molvm \
  --role reader \
  --scope $scope
```

4. Wenden Sie Berechtigungen auf den Azure Key Vault an, der Zugriff auf den Dienstprinzipal für die verwaltete Identität gewährt. Sie können dies über das Portal unter der Ressource Zugriffsrichtlinien für Key Vault tun, oder Sie können die Azure CLI verwenden. Verwenden wir die CLI, um zu sehen, wie die Informationen programmgesteuert abgerufen werden.

   Informieren Sie sich über das Azure AD Service Principal für Ihre verwaltete Identität. Filtern Sie nach dem `display-name` der VM, die Sie in Schritt 3 erstellt haben, z. B. `molvm`:

```
az ad sp list \
  --display-name molvm \
  --query [].servicePrincipalNames
```

Die Ausgabe ähnelt dem folgenden verkürzten Beispiel. Machen Sie sich keine Gedanken darüber, was diese Werte bedeuten. Sie müssen Ihnen hier nur die anfänglichen Berechtigungen zuweisen. Auch hier können Sie das Azure Portal verwenden, um die CLI zu vermeiden, wenn Sie sich damit nicht wohl fühlen.

Merken Sie sich den ersten `servicePrincipalName`. Dieser Wert wird verwendet, um Berechtigungen für Azure-Ressourcen wie Ihren Schlüsseltresor zu vergeben und wird im nächsten Schritt benötigt:

```
[
  "887e9665-3c7d-4142-b9a3-c3b3346cd2e2",
  "https://identity.azure.net//
  ➥ihxXtwZEiAeNXU8eED2Ki6FXRPkklthh84S60CiqA4="
]
```

5 Stellen Sie die Richtlinie für den Schlüsselspeicher so ein, dass der Service Principal für Ihre VM Geheimnisse lesen kann, und geben Sie den ersten `servicePrincipalName` aus Schritt 4 an:

```
az keyvault set-policy \
  --name azuremol \
  --secret-permissions get \
  --spn 887e9665-3c7d-4142-b9a3-c3b3346cd2e2
```

Ein Punkt, der hier zu beachten ist, ist dass, wenn die verwaltete Identität erstellt und auf die Ressourcengruppe übertragen wurde, dies nicht bedeutete, dass die VM dann alles tun konnte, was sie wollte. Erstens, die einzige Rolle, die für die Identität erstellt wurde, waren Leserechte für Ressourcen. Aber Sie mussten dem Schlüsseltresor selbst noch Berechtigungen zuweisen. Diese Sicherheits- und Berechtigungsebenen geben Ihnen eine detaillierte Kontrolle über die genauen Ressourcen, auf die jede Identität zugreifen kann.

Jetzt, da Sie Zugang zu einem Schlüsseltresor haben, wollen Sie wahrscheinlich wissen, wie Sie das Geheimnis abrufen können, nicht wahr?

15.3 Erhalten eines Geheimnisses aus einer VM mit einer Identität aus einem verwalteten Service

Sie haben ein Geheimnis in einem Schlüsseltresor für ein Datenbankpasswort gespeichert, und Sie haben eine VM mit einer verwalteten Identität, die den Zugriff zum Lesen dieses Geheimnisses aus dem Schlüsseltresor ermöglicht. Was nun? Wie holen Sie das Geheimnis heraus und nutzen es? Abbildung 15.5 zeigt, wie eine VM das IMDS verwendet, um den Zugriff auf eine Ressource, wie beispielsweise einen Schlüsseltresor, anzufordern. Lassen Sie uns die Schritte durchgehen, um zu sehen, wie die VM das Geheimnis abruft.

In den meisten Anwendungsfällen für Azure Key Vault gäbe es keine VM, die sich verbindet und die Geheimnisse auf diese Weise abruft. Key Vault punktet besonders, wenn Anwendungen selbst, innerhalb des Codes, Geheimnisse abrufen. Der Anwendungscode würde das entsprechende Azure-SDK verwenden, z. B. Python, .NET oder Java. Um komplexe Code Abstraktionen zu vermeiden, verwendet die folgende Übung eine VM und einige Befehlszeilenarbeit. Wenn Sie diese Übung durcharbeiten, denken Sie daran, dass diese Magie in der Regel im Code der Anwendung passieren würde.

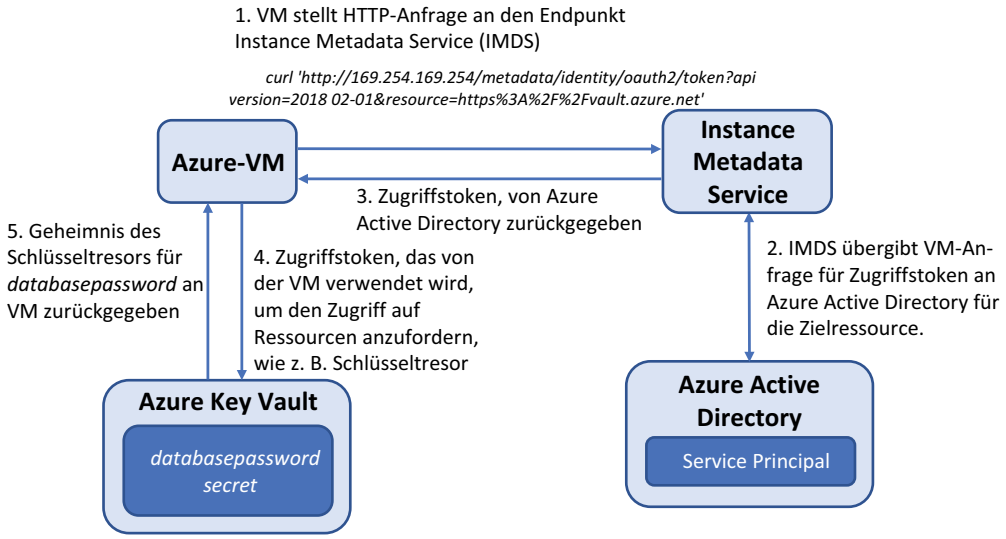


Abbildung 15.5 Die VM verwendet das IMDS, um den Zugriff auf einen Schlüsseltresor zu beantragen. Der Endpunkt kommuniziert mit Azure AD, um ein Zugriffstoken anzufordern. Das Zugriffstoken wird an die VM zurückgegeben, mit der dann der Zugriff vom Schlüsselspeicher angefordert wird. Wenn der Zugriff über den Schlüsselspeicher gewährt wird, wird das Geheimnis für `databasepassword` wird an die VM zurückgegeben.

Jetzt testen

Um ein Geheimnis auf einer VM mit einer verwalteten Identität abzurufen und zu verwenden, führen Sie die folgenden Schritte aus:

1. Ermitteln Sie die öffentliche IP-Adresse der VM, die Sie in der vorherigen Übung erstellt haben, z. B. `molvm`:

```
az vm show \
  --resource-group azuremolchapter15 \
  --name molvm \
  --show-details \
  --query [publicIps] \
  --output tsv
```

2. SSH für Ihre VM, wie z. B. `ssh azuremol@publicIps`.
3. Um auf einen Schlüsseltresor zuzugreifen, benötigen Sie ein Zugriffstoken. Dieses Zugriffstoken wird vom IMDS angefordert. Es handelt sich um eine HTTP-Anfrage. Auf einer Linux-VM können Sie das `curl`-Programm verwenden, um die Anfrage zu stellen. Das IMDS leitet Ihre Anfrage an AAD weiter:

```
curl 'http://169.254.169.254/metadata/identity/oauth2/token?
➥api-version=2018-02-01&resource=https%3A%2F%2Fvault.azure.net'
➥-H Metadata:true
```

4 Die Ausgabe ist etwas schwer zu lesen, da sie wie Textchaos aussieht. Sie ist im JSON Web Token (JWT)-Format. Um die JSON-Ausgabe zu verarbeiten und die Dinge besser lesbar für Menschen zu machen, installieren Sie einen JSON-Parser namens `jq`:

```
sudo apt-get update && sudo apt-get -y install jq
```

5 Stellen Sie Ihre `curl`-Anfrage erneut, aber diesmal sehen Sie sich die Ausgabe mit `jq` an:

```
curl 'http://169.254.169.254/metadata/identity/oauth2/token?
➥api-version=2018-02-01&resource=https%3A%2F%2Fvault.azure.net'
➥-H Metadata:true --silent | jq
```

Diese ersten Schritte sind für Sie, um zu sehen, wie die Anforderungen gestellt werden und wie die Ausgabe aussieht. Sie finden sie auch in Abbildung 15.6 dargestellt. Wenn Sie sich immer noch in der VM anmelden und manuell ein Zugriffstoken anfordern, was bringt es dann, eine verwaltete Identität zu verwenden? Sie könnten einfach ihre eigenen Zugangsdaten angeben. Beim Einsatz in der Produktion würden Sie wahrscheinlich ein Skript verwenden, das auf der VM ausgeführt wird, um die Anforderung eines Zugriffstokens automatisch durchzuführen und dann das Geheimnis aus dem Schlüsseltresor abzurufen. Machen wir weiter und sehen, wie Sie diesen Prozess automatisieren und das Geheimnis abrufen.

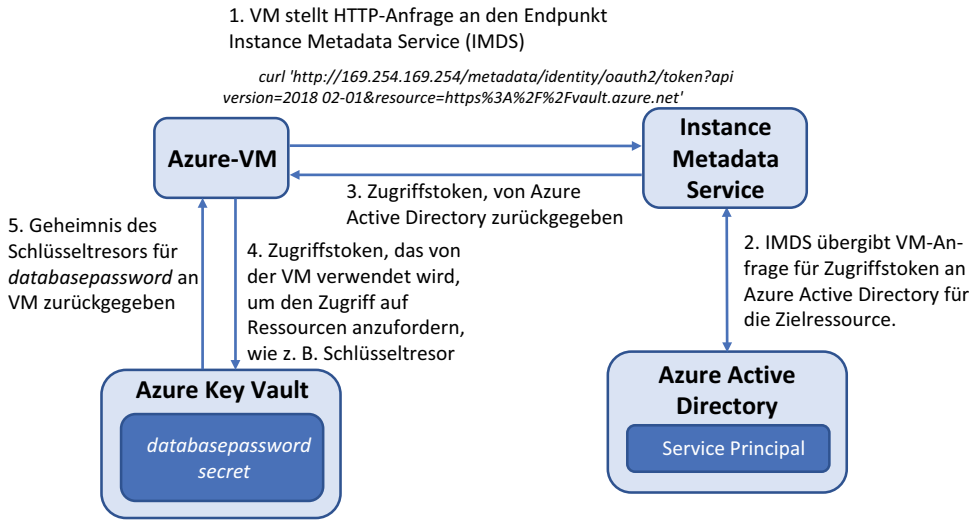


Abbildung 15.6 Die `curl`-Anfrage umfasst die ersten drei Schritte in diesem Diagramm. Die `curl`-Anfrage ist gestellt, der Endpunkt kommuniziert mit Azure AD und es wird ein Zugriffstoken ausgegeben.

6 Um die Dinge einfacher zu machen – und wenn Sie das alles in einem Skript machen würden – können Sie `jq` verwenden, um die curl-Antwort zu verarbeiten, dann einfach nur das Zugriffstoken extrahieren und es als Variable namens `access_token` einsetzen:

```
access_token=$(curl
➥'http://169.254.169.254/metadata/identity/oauth2/token?
➥api-version=2018-02-01&resource=https%3A%2F%2Fvault.azure.net'
➥-H Metadata:true --silent | jq -r '.access_token')
```

7 Als manueller Schritt, um zu verstehen, wie das aussieht, sehen Sie sich die `access_token`-Variabel an:

```
echo $access_token
```

8 Jetzt kommt der unterhaltsame Teil! Verwenden Sie das Zugriffstoken, um Ihr Geheimnis aus dem Schlüsseltresor anzufordern. Lassen Sie uns dies zunächst manuell tun, damit Sie verstehen, was passiert.

9 Holen Sie sich das Geheimnis mit einer weiteren `curl`-Anfrage, und formatieren Sie die Ausgabe mit `jq`: Geben Sie Ihren eigenen Namen des Schlüsseltresors ein, am Anfang von https:// address:

```
curl https://azuremol.vault.azure.net/secrets/databasepassword?
➥api-version=2016-10-01 -H "Authorization: Bearer $access_token"
➥--silent | jq
```

Die Ausgabe ähnelt der Folgenden, die den Wert des im Geheimnisses gespeicherten Passworts sowie einige zusätzliche Metadaten über das Geheimnis anzeigt, um die Sie sich momentan keine Gedanken machen müssen:

```
{
  "value": "SecureP@ssw0rd!",
  "contentType": "Database password",
  "id":
➥"https://azuremol.vault.azure.net/secrets/databasepassword/
➥87e79e35f57b41fdb882c367b5c1ffb3",
}
```

Diese `curl`-Anfrage ist der zweite Teil des Workflows, wie in Abbildung 15.7 dargestellt.

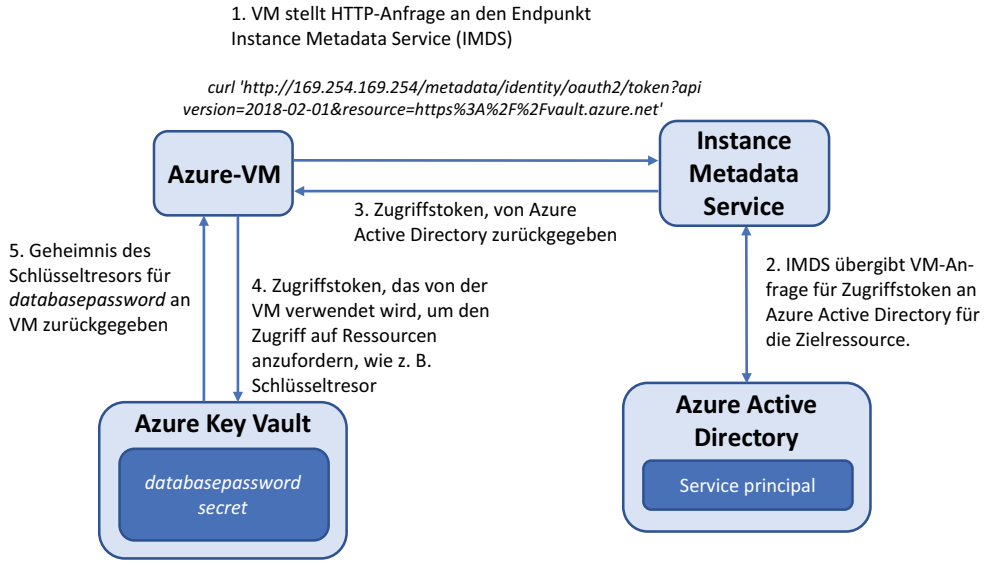


Abbildung 15.7 Diese zweite `curl`-Anfrage umfasst die letzten beiden Schritte im Diagramm. Das Zugriffstoken wird verwendet, um das Geheimnis aus dem Schlüsseltresor anzufordern. Die JSON-Antwort wird zurückgegeben und enthält den Wert des Geheimnisses.

10 So wie Sie eine Variable zum Speichern des Zugriffstokens verwendet haben, können Sie in einem Skript den Wert des Geheimnisses auch einer Variablen zuweisen. Verwenden Sie diesmal `jq`, um die Antwort zu verarbeiten, extrahieren Sie nur das Wertgeheimnis und setzen Sie es als Variable mit Namen `database_password` ein:

```
database_password=$(curl
➥https://azuremol.vault.azure.net/secrets/databasepassword?
➥api-version=2016-10-01 -H "Authorization: Bearer $access_token"
➥--silent | jq -r '.value')
```

11 Auch hier, als manueller Schritt zum besseren Verständnis des Prozesses, sollten Sie sich den Inhalt der Variable `database_password` ansehen:

```
echo $database_password
```

Hoffentlich sind Sie noch mit dabei! Wenn Sie beispielsweise eine Anwendung in Python, ASP.NET oder Node.js schreiben, ist der Prozess ähnlich, wenn Sie eine Anforderung für das Zugriffstoken stellen und dann mit dem Token ein Geheimnis aus einem Schlüsselspeicher anfordern. Es gibt sicher andere Bibliotheken, die Sie in Ihrem Code verwenden können, anstatt `jq` aus der Kommandozeile heraus zu benutzen.

Für eine kurze Übersicht können wir alle diese Schritte in zwei Zeilen zusammenfassen, wie in der folgenden Auflistung.

Auflistung 15.1 Anfordern eines Zugriffstokens und dann eines Geheimnisses aus einem Schlüsseltresor

```
access_token=$(curl
➥'http://169.254.169.254/metadata/identity/oauth2/token?
➥api-version=2018-02-01&resource=https%3A%2F%2Fvault.azure.net'
➥-H Metadata:true --silent | jq -r '.access_token')
database_password=$(curl
➥https://azuremol.vault.azure.net/secrets/databasepassword?
➥api-version=2016-10-01 -H "Authorization: Bearer $access_token"
➥-silent | jq -r '.value')
```

Was nun? Die verwaltete Identität ihrer VM kann ein Geheimnis aus einem Schlüsseltresor abrufen. Sehen wir uns an, wie Sie mit dieser verwalteten Identität MySQL Server installieren und konfigurieren können.

Bei Ubuntu können Sie Konfigurationsoptionen für Paket-Installer wie MySQL Server festlegen. Mit diesen Konfigurationsoptionen können Sie Werte wie Benutzungsnamen und Kennwörter eingeben und diese automatisch für den entsprechenden Teil des Installationsprozesses verwenden. Die manuellen Aufforderungen zur Eingabe eines Passworts, wie Sie sie vielleicht schon in Kapitel 2 gesehen haben, sind verschwunden.

12 Legen Sie die Konfigurationsoptionen für die MySQL Server-Passwörter mit der Variable `database_password` fest, die Sie in Schritt 10 angelegt haben:

```
sudo debconf-set-selections <<< "mysql-server mysql-server/root_password
➥password $database_password"
sudo debconf-set-selections <<< "mysql-server mysql-
➥server/root_password_again password $database_password"
```

13 Installieren Sie MySQL Server. Es gibt keine Eingabeaufforderungen, da das Kennwort durch Konfigurationsabfragen bereitgestellt wird:

```
sudo apt-get -y install mysql-server
```

14 Lasst uns beweisen, dass all das funktioniert hat! Sehen Sie sich die `database_password`-Variable an, damit klar wird, wie Ihr Passwort aussehen sollte:

```
echo $database_password
```

15 Melden Sie sich beim MySQL Server an. Wenn Sie nach einem Passwort gefragt werden, geben Sie den Wert von `database_password` ein. Dies ist der Wert des Geheimnisses aus dem Schlüsseltresor:

```
mysql -u root -p
```

Sie sind im MySQL Server angemeldet, der bestätigt, dass das Geheimnis aus dem Schlüsselspeicher verwendet wurde, um die SQL Server-Anmeldeinformationen erfolgreich zu erstellen!

16 Geben Sie zweimal `exit` ein, um die MySQL Server-Eingabeaufforderung zu schließen, und schließen Sie dann Ihre SSH-Sitzung für die VM.

Dies war ein einfaches Beispiel. Sie müssten immer noch den MySQL Server sichern und zusätzliche Anmeldeinformationen für Anwendungen bereitstellen, die beispielsweise auf Datenbanken oder Tabellen zugreifen. Der Vorteil der Verwendung eines Geheimnisses aus einem Schlüsseltresor ist, dass Sie sicherstellen, dass alle Passwörter gleich sind. Wenn Sie beispielsweise Skalierungsgruppen für virtuelle Maschinen verwenden, dann kann jede VM-Instanz automatisch das Geheimnis anfordern und den MySQL Server installieren, damit er für die Bereitstellung Ihrer Anwendungsdaten bereit ist. Diese Passwörter werden nie in Skripten definiert, und niemand muss sehen, wie die Kennwörter lauten. Sie könnten sogar zufällige Kennwörter generieren und diese als Geheimnisse in einem Schlüsseltresor rotieren lassen.

Das Speichern von Kennwörtern in einem Schlüsseltresor ist super, aber können Sie einen Schlüsseltresor verwenden, um Zertifikate zu speichern und Sie automatisch aus Ihren Anwendungen oder VMs abzurufen? Natürlich können Sie das!

15.4 Erstellen und Einspeisen von Zertifikaten

Digitale Zertifikate sind eine gängige Form der Sicherheit und Authentifizierung in Webservices und Anwendungen. Zertifikate werden von einer Zertifizierungsstelle (CA) ausgestellt, der die Anwender*innen (hoffentlich!) vertrauen. Mit dem Zertifikat können Benutzer*innen überprüfen, ob eine Website oder Anwendung tatsächlich die ist, für die sie sich ausgibt. Jedes Mal, wenn Sie eine Website mit einer Webbrowser-Adresse sehen, die mit https:// beginnt und ein Schlosssymbol hat, wird der Datenverkehr verschlüsselt und durch ein digitales Zertifikat gesichert.

Die Verwaltung digitaler Zertifikate kann oft zu einer großen Verwaltungsaufgabe werden. Ein häufiges Problem ist, wie man Zertifikate speichert und Zugriff auf sie gewährt, wenn Dienste und Anwendungen sie benötigen. In den vorangegangenen Übungen haben wir untersucht, wie ein Schlüsseltresor zum Teilen sicherer Geheimnisse und Schlüssel mit Diensten und Anwendungen eingesetzt werden kann. Allerdings kann ein Schlüsseltresor dasselbe auch mit Zertifikaten tun. Wie in Abbildung 15.8 dargestellt, kann ein Schlüsseltresor zum Anfordern, Ausstellen und Speichern von Zertifikaten verwendet werden.

Im Produktionseinsatz sollten Sie immer eine vertrauenswürdige CA verwenden, um Ihre Zertifikate auszustellen. Für den internen Gebrauch können Sie selbstsignierte Zertifikate ausstellen, die Sie selbst erstellen. Diese

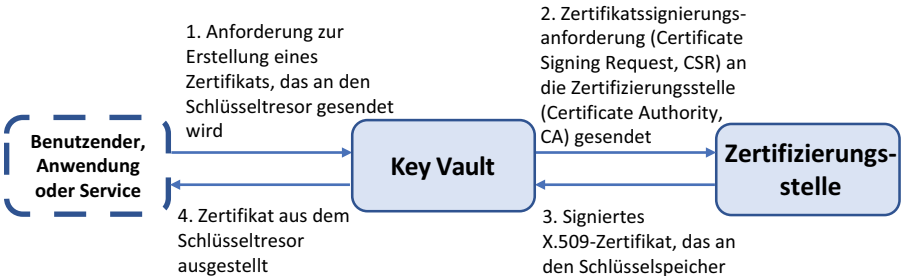


Abbildung 15.8 Ein*e Benutzer*in, eine Anwendung oder ein Dienst kann ein neues Zertifikat von einem Schlüsselspeicher anfordern. Eine Zertifikatssignierungsanforderung (CSR) wird vom Schlüsselspeicher an eine Zertifizierungsstelle (CA) gesendet. Azure Key Vault kann auch als eigene CA fungieren, um selbstsignierte Zertifikate zu generieren. Die CA stellt dann ein signiertes X.509-Zertifikat aus, das im Schlüsseltresor gespeichert ist. Schließlich gibt der Schlüsseltresor das Zertifikat an den ursprünglichen Antragsteller zurück.

Diese selbstsignierten Zertifikate werden von anderen Diensten und Anwendungen nicht als vertrauenswürdig eingestuft, sodass sie in der Regel eine Warnung erzeugen. Allerdings ermöglichen Ihnen selbstsignierte Zertifikate auch, schnell loszulegen und sicherzustellen, dass Ihr Code mit verschlüsseltem Datenverkehr wie erwartet funktioniert.

Azure Key Vault kann für Sie selbstsignierte Zertifikate generieren. Tiefergreifend betrachtet fungieren Schlüsseltresore als eigene CA, um Zertifikate anzufordern, auszustellen und dann zu speichern. Verwenden wir diese Möglichkeit, ein selbstsigniertes Zertifikat zu generieren und sehen uns an, wie es einfach in eine VM einfügt werden kann. Das Zertifikat wird dann für einen einfachen Webserver verwendet, um Ihnen zu zeigen, wie Sie SSL schnell aktivieren können, um Ihren Webverkehr zu schützen.

Jetzt testen

Um ein Zertifikat zu erstellen und in eine VM einzubinden, führen Sie die folgenden Schritte aus:

1. Erstellen Sie ein selbstsigniertes Zertifikat in Azure Key Vault und geben Sie einen Namen ein, z. B. `molcert`. Richtlinien werden verwendet, um Eigenschaften wie Ablaufzeiten, Verschlüsselungsstärke und Zertifikatsformat zu definieren. Sie können verschiedene Richtlinien erstellen, um den Anforderungen Ihrer Anwendungen und Dienste gerecht zu werden. Verwenden Sie für diese Übung die Standardrichtlinie, die ein 2.048-Bit-Zertifikat erstellt und ein Jahr lang gültig ist:

```
az keyvault certificate create \
  --vault-name azuremol \
  --name molcert \
  --policy "$(az keyvault certificate get-default-policy)"
```

2. Um das Zertifikat in Aktion zu sehen, erstellen Sie eine weitere VM, wie z. B. `molwinvm`. Erstellen Sie diesmal eine Windows-VM, die Windows Server 20119 verwendet, sodass Sie Werbung für das Betriebssystem verbreiten und sehen können, dass diese Key Vault-Features nicht von einem bestimmten Betriebssystem abhängen! Geben Sie Ihren eigenen Admin-Benutzungsnamen und Ihr eigenes Kennwort an:

```
az vm create \
  --resource-group azuremolchapter15 \
  --name molwinvm \
  --image win2019datacenter \
  --admin-username azuremol \
  --admin-password P@ssw0rd1234
```

3 Sie können das Zertifikat direkt aus Azure CLI heraus automatisch zur VM hinzufügen. Dieser Ansatz verlässt sich nicht auf eine verwaltete Identität. Die Azure-Plattform injiziert das Zertifikat mithilfe des Windows Azure VM-Agents.

Fügen Sie Ihr Zertifikat, wie z. B. `molcert`, zu der VM hinzu, die Sie im vorherigen Schritt erstellt haben, z. B. `molwinvm`:

```
az vm secret add \
  --resource-group azuremolchapter15 \
  --name molwinvm \
  --keyvault azuremol \
  --certificate molcert
```

4 Verbinden Sie sich mit der VM, und überprüfen Sie, ob das Zertifikat korrekt eingespeist wurde. Um sich mit Ihrer VM zu verbinden, holen Sie sich zunächst deren öffentliche IP-Adresse:

```
az vm show \
  --resource-group azuremolchapter15 \
  --name molwinvm \
  --show-details \
  --query [publicIps] \
  --output tsv
```

Verwenden Sie einen lokalen Microsoft-Remotedesktopclient auf Ihrem Computer, um eine Verbindung mit Ihrer VM herzustellen. Verwenden Sie die Anmeldeinformationen für die Verbindung mit localhost\azuremol, nicht die Standard-Anmeldeinformationen Ihres lokalen Computers, die Ihr Remotedesktopclient verwenden kann, wie in Abbildung 15.9 dargestellt.

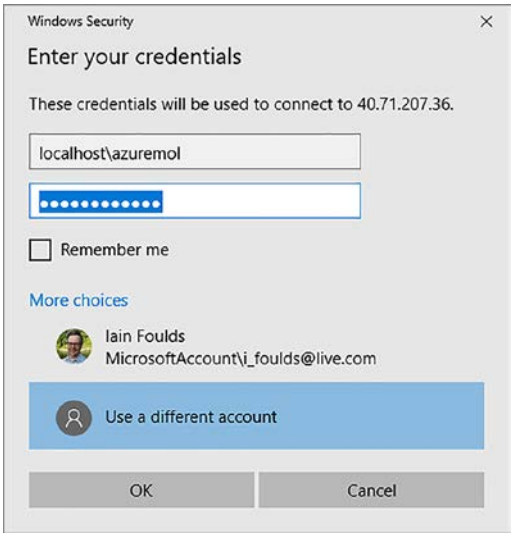


Abbildung 15.9 Ihr Remote Desktop Client kann versuchen, Ihre standardmäßigen lokalen Computer-Anmeldeinformationen zu verwenden. Wählen Sie stattdessen „Use a Different Account“ (Ein anderes Konto verwenden), und geben Sie dann die localhost\azuremol-Anmeldeinformationen an, die Sie beim Erstellen der VM angegeben haben.

5 Nachdem Sie angemeldet sind, wählen Sie die Windows-Starttaste, und geben Sie dann `mmc` ein und öffnen Sie die Microsoft Management Console.

6 Wählen Sie „File“ (Datei) > „Add / Remove Snap-in“ (Snap-In hinzufügen/entfernen), und fügen Sie dann das Snap-In für Zertifikate hinzu.

7 Fügen Sie Zertifikate für das Computerkonto hinzu, wählen Sie „Next“ (Weiter) und dann „Finish“ (Fertig stellen).

8 Wählen Sie OK, um das Fenster „Add / Remove Snap-in“ (Snap-In hinzufügen/entfernen) zu schließen.

9 Erweitern Sie den Ordner „Certificates (Local Computer) > Personal > Certificates“ (Zertifikate (Lokaler Computer) > Persönlich > Zertifikate). Das Zertifikat von Azure Key Vault, das Sie in die VM injiziert haben, wird aufgelistet, wie beispielsweise CLIGetDefaultPolicy, wie in Abbildung 15.10 dargestellt.

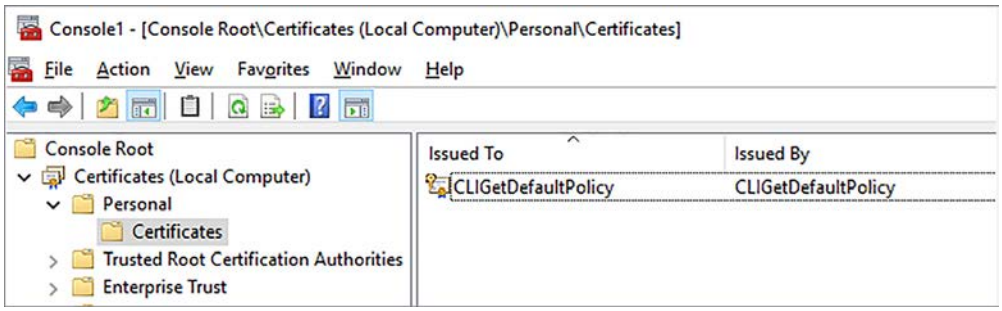


Abbildung 15.10 Fügen Sie in der Microsoft Management Console das Snap-In für Zertifikate auf dem lokalen Computer hinzu. Erweitern Sie den Speicher für persönliche Zertifikate, um installierte Zertifikate anzuzeigen. Das vom Schlüsseltresor eingespeiste Zertifikat ist aufgelistet.

Das ist der ganze Trick! Erstellen Sie das Zertifikat im Schlüsseltresor, und fügen Sie das Zertifikat dann der VM hinzu. Das Zertifikat wird im lokalen Zertifikatsspeicher des Computers abgelegt, sodass jeder Dienst oder jede Anwendung darauf zugreifen kann. Auf einer Windows-VM werden die Zertifikate im lokalen Zertifikatscache gespeichert, wie in dieser Übung gezeigt. Auf Linux-VMs werden .prv- und.crt-Dateien für den privaten und öffentlichen Teil des Zertifikats in /var/lib/waagent/ gespeichert. Sie können die Zertifikate an einen beliebigen Ort für Ihre Anwendung oder Ihren Dienst verschieben.

Zertifikate können für die Authentifizierung zwischen Clients und Servern oder zwischen Anwendungskomponenten und Diensten verwendet werden. Ein gängiges Beispiel ist, dass ein Webserver ein SSL-Zertifikat verwendet, was Sie auch in der Übung am Ende des Kapitels tun.

15.5 Übung: Konfigurieren eines sicheren Webservers

In der letzten Übung haben Sie ein selbstsigniertes Zertifikat von Azure Key Vault in eine Windows-VM eingespeist. Installieren und konfigurieren Sie für diese Übung den IIS-Webserver, der das Zertifikat verwenden soll. Folgen Sie dabei dieser Anleitung:

1. Öffnen Sie PowerShell auf Ihrer Windows-VM, und installieren Sie dann den IIS-Webserver:

   ```
   Add-WindowsFeature Web-Server -IncludeManagementTools
   ```

2. Öffnen Sie den Internet Information Server (IIS) Manager. Sie können dies über das Menü „Tools“ (Extras) im Server Manager tun.
3. Wählen Sie für die Standardwebsite die Option „Edit Bindings“ (Bindungen bearbeiten).
4. Fügen Sie eine HTTPS-Bindung für alle nicht zugewiesenen IP-Adressen auf Port 443 hinzu.
5. Wählen Sie das selbstsignierte Zertifikat, das Sie aus dem Schlüsseltresor erstellt und injiziert haben, in der Regel so etwas wie CLIGetDefaultPolicy.
6. Öffnen Sie einen Webbrowser auf der VM und geben Sie `https://localhost` ein. Sie haben im Schlüsseltresor ein selbstsigniertes Zertifikat generiert, sodass der Webbrowser ihm nicht traut.
7. Akzeptieren Sie die Warnung, um fortzufahren, und vergewissern Sie sich, dass das HTTPS-Binding funktioniert.
8. Zurück in der Azure Cloud Shell oder im Portal erstellen Sie eine NSG-Regel für die VM auf dem TCP-Port 443. Geben Sie `https://yourpublicipaddress` in einem Webbrowser auf Ihrem lokalen Computer ein. Dies ist die Erfahrung, die Ihre Benutzer*innen erhalten würden, mit einer Warnung vor einem nicht vertrauenswürdigen, selbstsignierten Zertifikat. Denken Sie in den meisten Anwendungsfällen daran, eine vertrauenswürdige interne oder externe Zertifizierungsstelle zu verwenden, um vertrauenswürdige Zertifikate zu generieren und diese in einem Schlüsseltresor zu speichern.

16 Azure Security Center und Updates

Wäre es nicht toll, wenn Azure intelligent genug war, um alle Ihre wichtigsten Anwendungsressourcen zu überwachen und Sie über Sicherheitsprobleme zu informieren? Oder was, wenn Ihr Unternehmen bereits Sicherheitsrichtlinien definiert hat? (wenn Sie keine Sicherheitsrichtlinien haben, stoppen Sie bitte sofort und notieren Sie sich, dass Sie sie brauchen!) Wie stellen Sie im letzteren Fall sicher, dass Ihre Azure-Bereitstellungen konform bleiben? Wenn Sie jemals eine IT-Sicherheitsüberwachung mitgemacht haben, dann wissen Sie, wie viel Spaß es machen kann, sich eine Liste von Fehlkonfigurationen in Ihrer Umgebung anzusehen, besonders, wenn es um grundlegenden Sicherheitslücken geht, die Sie hätten kennen müssen!

Das Azure Security Center bietet einen zentralen Ort für Sicherheitswarnungen und Empfehlungen, die für Ihre Überprüfung gruppiert werden können. Sie können Ihre eigenen Sicherheitsrichtlinien definieren und dann Azure den Status Ihrer Ressourcen auf Konformität überwachen lassen.

In diesem Kapitel wird erläutert, wie das Security Center Sie auf Probleme aufmerksam machen und Schritte zu deren Behebung bereitstellen kann, wie Sie den zeitlich begrenzten VM-Zugriff zur Steuerung und Überprüfung von Remote-Verbindungen nutzen können und wie das Update-Management Ihre VMs automatisch mit den neuesten Sicherheitspatches auf dem neuesten Stand hält.

16.1 Azure Security Center

Überall in diesem Buch sind wir auf sicherheitsrelevante Themen eingegangen, wie das Erstellen und Konfigurieren von Netzwerksicherheitsgruppen (NSGs), um den Zugriff auf VMs zu beschränken, und wie nur verschlüsselter Datenverkehr für Azure Storage-Konten zugelassen wird. Woher wissen Sie nun, wo Sie anfangen sollen? Und wie können Sie überprüfen, ob Sie alle empfohlenen Sicherheitsmaßnahmen angewendet haben? Hier kann das Azure Security Center helfen. Es überprüft Ihre Umgebung auf Bereiche, die Sie möglicherweise übersehen haben.

Das Azure Security Center scannt Ihre Ressourcen, empfiehlt Korrekturen und hilft bei der Behebung von Sicherheitsproblemen, wie in Abbildung 16.1 dargestellt. Wenn Sie nur ein paar Test-VMs und ein einzelnes virtuelles Netzwerk in Ihrem Azure-Abonnement haben, mag es nicht so schwer erscheinen, den Überblick zu behalten, welche Sicherheitseinschränkungen Sie einführen müssen. Aber wenn Sie sich auf zehn, hundert oder sogar tausend VMs vergrößern, wird es nicht mehr möglich, manuell zu verfolgen, welche Sicherheitskonfigurationen auf jede VM angewendet werden müssen.

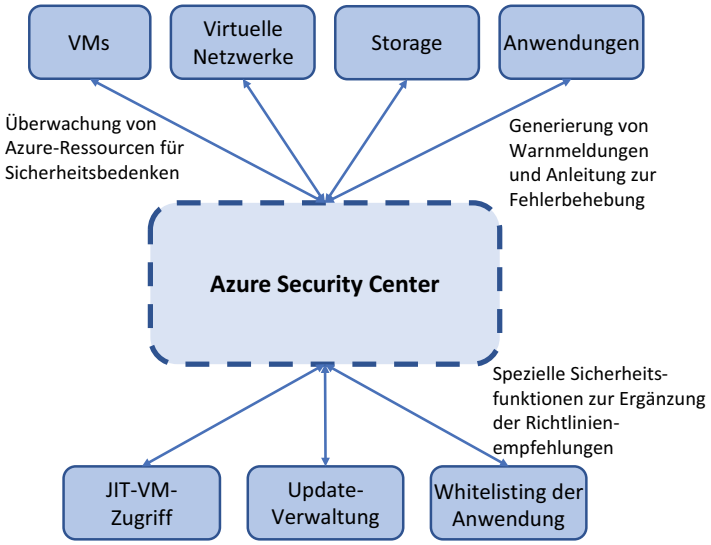


Abbildung 16.1 Azure Security Center überwacht Ihre Azure-Ressourcen und verwendet definierte Sicherheitsrichtlinien, um Sie über potenzielle Bedrohungen und Schwachstellen zu informieren. Sie erhalten Empfehlungen und Schritte zur Behebung von Problemen. Sie können auch einen zeitlich begrenzten VM-Zugriff verwenden, Sicherheitsupdates überwachen und anwenden sowie Anwendungen auf der Whitelist steuern, die auf VMs ausgeführt werden können.

Das Security Center kann Sie auch über allgemeine, empfohlene Maßnahmen informieren, z. B. wenn eine VM keine Diagnose aktiviert hat. Erinnern Sie sich an Kapitel 12, als wir uns mit der Überwachung und Fehlerbehebung von VMs befasst haben? Sie müssen den Diagnose-Agent installieren und konfigurieren, *bevor* ein Problem auftritt. Wenn Sie einen Sicherheitsverstoß vermuten, können Sie möglicherweise schon nicht mehr auf die VM- und Review-Log-Dateien zugreifen. Aber wenn Sie die Diagnose-Erweiterung zum Streamen von Log-Dateien an Azure Storage konfiguriert hätten, könnten Sie überprüfen, was passiert ist, und die Ursache und das Ausmaß des Problems aufspüren (wenn alles gut geht).

Jetzt testen

Um mit dem Azure Security Center zu beginnen, führen Sie die folgenden Schritte aus:

1. Öffnen Sie das Azure Portal, und wählen Sie das Cloud Shell-Symbol aus dem oberen Menü.
2. Erstellen Sie eine Ressourcengruppe und geben Sie einen Namen, wie `azuremolchapter16`, und einen Standort, wie `eastus`, ein:

```
az group create --name azuremolchapter16 --location eastus
```

3 Erstellen Sie eine einfache Linux-VM, damit das Security Center etwas zu überwachen hat und Empfehlungen geben kann:

```
az vm create \
  --resource-group azuremolchapter16 \
  --name azuremol \
  --image ubuntults \
  --admin-username azuremol \
  --generate-ssh-keys
```

4 Sobald die VM bereitgestellt ist, schließen Sie Cloud Shell.

5 Wählen Sie im Azure Portal aus der Liste der Dienste links die Option Security Center. Beim ersten Öffnen des Dashboards dauert es einige Sekunden, bis alle verfügbaren Komponenten vorbereitet sind, siehe Abbildung 16.2.

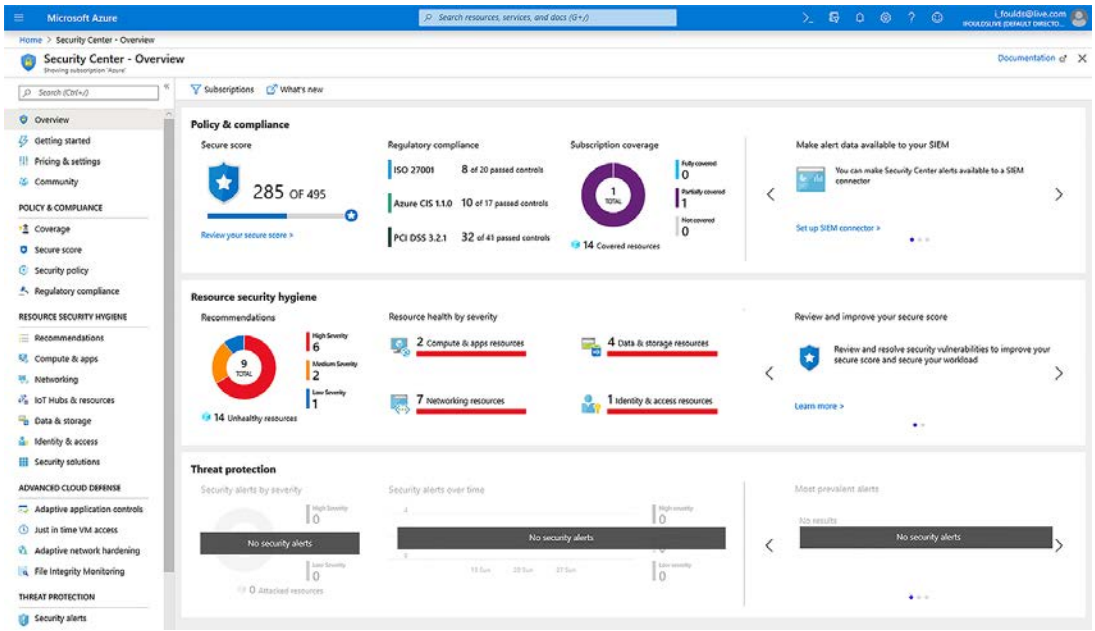


Abbildung 16.2 Das Azure Security Center Overview-Fenster bietet eine Liste von Empfehlungen, Warnungen und Ereignissen. Sie können einen Haupt-Ressourcentyp wie „Compute“ oder „Networking“ auswählen, um eine Liste von Sicherheitselementen anzuzeigen, die für diese Ressourcen spezifisch sind.

Das Security Center untersucht, wie Ressourcen wie VMs, NSG-Regeln und Speicher bereitgestellt werden. Eingebaute Sicherheitsgrundlagen werden verwendet, um Probleme zu identifizieren und Empfehlungen zu geben. Beispielsweise erzeugt das virtuelle Netzwerk, das Sie mit Ihrer VM eingesetzt haben, ein paar Warnungen, wie in Abbildung 16.3 dargestellt. Sie können und sollten Ihre eigenen Sicherheitsrichtlinien implementieren, die Azure mitteilen, wie Sie den Zugriff einschränken möchten oder was getan werden muss, um Geschäftsanforderungen zu erfüllen. Während Sie dann Ressourcen erstellen oder aktualisieren, überwacht Azure

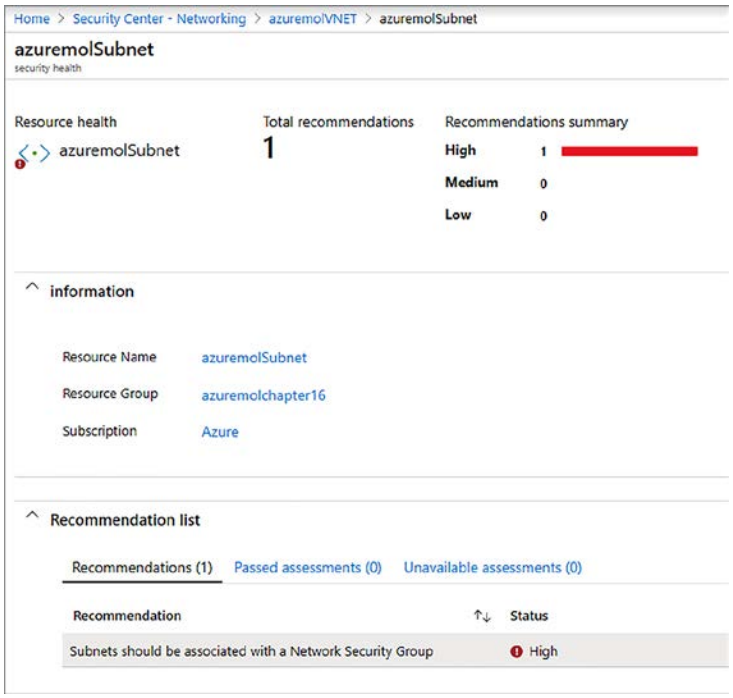


Abbildung 16.3 Das virtuelle Netzwerk für Ihre VM löst bereits Sicherheitswarnungen aus. In diesem Beispiel wird gewarnt, dass eine Netzwerksicherheitsgruppe dem Subnetz zugeordnet werden sollte.

kontinuierlich, ob es Abweichungen von diesen Richtlinien gibt, und warnt Sie darüber, welche Schritte unternommen werden müssen, um die Sicherheitsprobleme zu beheben. In diesem Kapitel verwenden Sie die standardmäßigen Azure-Sicherheitsrichtlinien in diesem Kapitel, können aber schon einmal an bestimmte Sicherheitskonfigurationen denken, die Sie auf Ihre VMs anwenden möchten, und wie sie in Ihren eigenen benutzungsdefinierten Richtlinien definiert werden könnten.

6 Wählen Sie Compute & Apps „Computing und Apps) aus dem linken Menü im Security Center-Fenster und dann „VMs“ und „Computers“ (Computer) aus.

7 Wählen Sie die VM, die Sie in Schritt 3 erstellt haben. Obwohl Sie diese VM gerade erstellt und Standardwerte aus der Azure CLI verwendet haben, werden einige Sicherheitswarnungen angezeigt:

Erkunden Sie einige dieser Empfehlungen. Wenn Sie die einzelnen Empfehlungen auswählen, erhalten Sie von einigen nur weitere Informationen, andere führen Sie durch Korrekturmaßnahmen. Dies sind keine knallharten Regeln, sondern Empfehlungen und Best Practices. In ihrer eigenen Umgebung ergeben einige davon möglicherweise keinen Sinn. Aber sie sind ein guter Ausgangspunkt, um zu erfahren, was Sie tun sollten, um Ressourcen zu schützen, während Sie sie in Azure erstellen.

16.2 Zeitlich beschränkter Zugriff

In Abschnitt 16.1 haben Sie erfahren, wie das Security Center vorschlägt, den Umfang der eingehenden Remote-Konnektivität einzuschränken. Sie könnten einen IP-Bereich bereitstellen, um den Datenverkehr zu begrenzen, aber im Idealfall öffnen Sie die Inbound-Connectivity nur dann, wenn sie benötigt wird. Auf diese Weise ist die VM für Remote-Verbindungen vollständig geschlossen und bei Bedarf nur für kurze Zeit

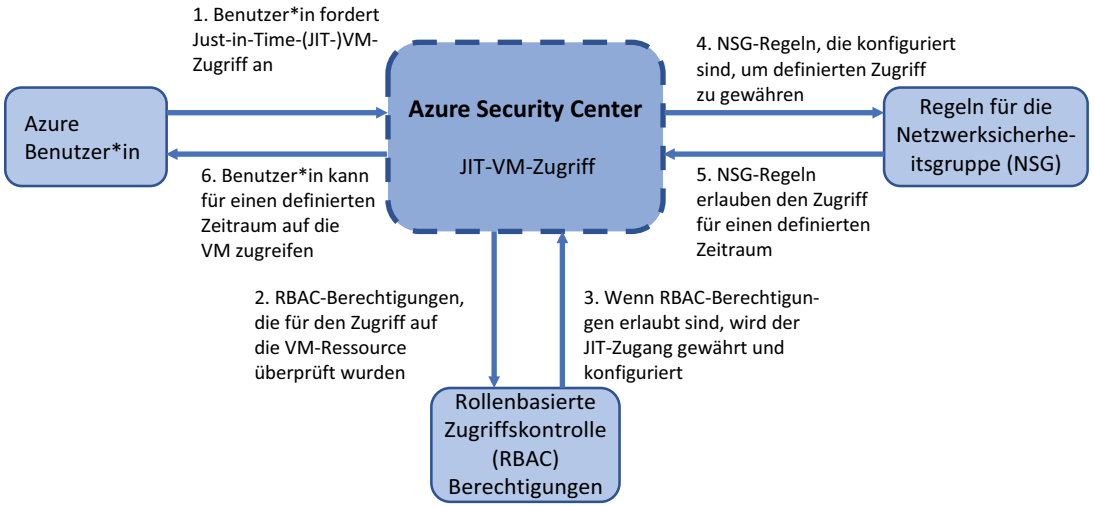


Abbildung 6.4 Beim JIT-VM-Zugriff sind NSG-Regeln konfiguriert, um Remote-Verbindungen mit einer VM zu verweigern. RBAC-Berechtigungen werden verwendet, um Berechtigungen zu überprüfen, wenn ein Benutzender den Zugriff auf eine VM anfordert. Diese Anfragen werden geprüft, und wenn der Anfrage stattgegeben wird, werden die NSG-Regeln aktualisiert, um den Datenverkehr aus einem bestimmten IP-Bereich für einen definierten Zeitraum zu ermöglichen. Der Benutzende kann nur während dieser Zeit auf die VM zugreifen. Nach Ablauf der Zeit kehren die NSG-Regeln automatisch in den deny-Zustand zurück.

zugänglich. Und ja, Sie sollten dieses kurze Zeitfenster der Konnektivität immer noch auf einen bestimmten IP-Bereich beschränken! Hier kommt der zeitlich begrenzte (Just-in-Time-(JIT-))VM-Zugriff zum Tragen, wie in Abbildung 16.4 dargestellt.

Beim JIT-Zugriff passt Security Center die Zugriffsbeschränkungen für eine VM dynamisch an. Wenn sie aktiviert ist, werden NSG-Regeln erstellt, die den gesamten Remoteverbindungsverkehr verweigern. Ein*e Benutzer*in kann dann nur bei Bedarf Zugriff auf eine VM anfordern. In Kombination mit rollenbasierter Zugriffssteuerung (siehe Kapitel 6) bestimmt Security Center, ob ein*e Benutzer*in bei der Anforderung einer Verbindung Zugriffsrechte auf eine VM hat. Wenn der oder die Benutzer*in über Berechtigungen verfügt, aktualisiert das Security Center die relevanten NSG-Regeln, um eingehenden Datenverkehr zu ermöglichen. Diese Regeln werden nur für ein bestimmtes Zeitfenster angewendet. Sobald diese Zeit abgelaufen ist, werden die Regeln umgekehrt und die VM wird für Remote-Verbindungen wieder geschlossen. Wenn Sie über eine aktive Verbindung zu einer VM verfügen, wird die Verbindung nicht automatisch getrennt, wenn die Zeit abläuft. Sie können Ihre Wartungs- oder Fehlerbehebungsarbeiten beenden und die Verbindung trennen, wenn Sie fertig sind, aber Sie können keine neue Verbindung starten, es sei denn, Sie fordern den JIT-Zugriff erneut an.

Trinken aus einem Hydranten

Wir haben uns nicht wirklich mit Azure Firewall befasst. Es handelt sich dabei um eine virtuelle Netzwerkressource, die einer physischen On-Premises-Firewall ähnlicher ist, als die NSGs selbst. Wenn Sie mehr Flexibilität und Kontrolle über den Datenverkehr benötigen, ist Azure Firewall eine großartige Option, auch wenn damit Kosten verbunden sind.

Ohne zu tief auf Azure Firewall einzugehen, möchte ich darauf hinweisen, dass Azure Security Center auch in Azure Firewall integriert werden kann, um die erforderlichen Regeln zu öffnen und zu schließen. Wenn Sie Azure Firewall verwenden, um den

VM-Datenverkehr in virtuellen Netzwerken zu schützen, nicht nur NSGs, können Sie die automatisierte Regelverwaltung von JIT-VM-Zugriff weiter nutzen.

Weitere Informationen zu Azure Firewall finden Sie in der Dokumentation unter https://docs.microsoft.com/azure/firewall/overview.

Wann würden Sie JIT in Ihrem fiktiven Pizzageschäft einsetzen? Denken Sie an alle VMs, die Ihre Webanwendung, Ihr Bestellsystem oder Ihre Geschäftslogik-Anwendungen ausführen würden. Würden Sie wollen, dass diese mit dem Internet verbunden sind und für andere jederzeit zugänglich sind? Ich hoffe nicht! Es gibt triftige Gründe für den Remotezugriff mit SSH oder RDP, aber versuchen Sie immer zu beschränken, wie lange dieser Zugriff möglich ist. Selbst wenn Sie NSG-Regeln haben, die den Zugriff auf bestimmte IP-Bereiche einschränken, fügt JIT eine weitere Schutzebene für Azure-Benutzer*innen hinzu, und erstellt dann einen einfacheren Audit-Trail, über den das Security Center Berichte bereitstellen kann.

Jetzt testen

Um den JIT-VM-Zugriff zu aktivieren, führen Sie die folgenden Schritte aus:

1. Öffnen Sie das Azure Portal, und wählen Sie Security Center aus dem Menü links.
2. Wählen Sie unter „Advanced Cloud Defense“ (Erweiterte Cloud-Abwehr) die Option „Just in Time VM access“ (JIT-VM-Zugriff).
3. Wenn Sie dazu aufgefordert werden, wählen Sie die Option „Try Just in Time VM access“ (JIT-VM-Zugriff ausprobieren) oder „Upgrade to Standard Tier of Security Center“ (Führen Sie ein Upgrade auf den Tarif „Standard“ des Security Centers durch). Diese kostenlose Testversion dauert 60 Tage und verlängert sich nicht automatisch. Dies überschneidet sich mit Ihrem kostenlosen Azure-Konto und kostet Sie kein Geld. Wählen Sie die Option „Apply Standard Plan“ (Standardplan anwenden), und warten Sie dann einige Augenblicke, bis er aktiviert ist. Sobald er aktiviert ist, müssen Sie möglicherweise das Azure Portal schließen und wieder öffnen, bevor Sie die folgenden Schritte ausführen können.
4. Wählen Sie im Fenster Security Center erneut „Just in Time VM Access“ (JIT-VM-Zugriff) aus. Sobald Ihr Standardplan aktiviert ist, können Sie sich eine Liste der zu verwendenden VMs anzeigen lassen.
5. Wählen Sie Ihre VM aus, und wählen Sie dann „Zugriff anfordern“ (Request Access) (wie in Abbildung 16.5 dargestellt).

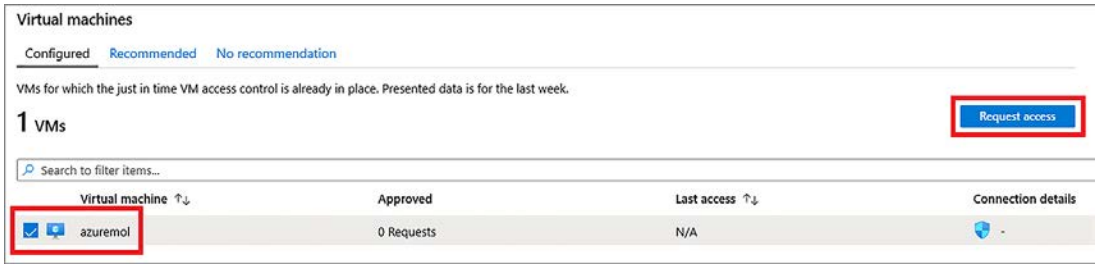


Abbildung 16.5 Wählen Sie eine VM aus den empfohlenen Optionen aus, und wählen Sie dann „Enable JIT on 1 VMs“ (JIT auf 1 VM aktivieren). Der Status zeigt jetzt an, dass diese VM für alle Remotezugriffe geöffnet ist, was den Schweregrad des Sicherheitsproblems als „Hoch“ kennzeichnet.

Standardmäßig definiert JIT Regeln, die Ports für SSH (Port 22), RDP (Port 3389) und PowerShell Remoting (Port 5985 und 5986) für einen Zeitraum von drei Stunden öffnen können.

6 Wählen Sie für diese Übung SSH aus ihrer eigenen IP-Adresse. Geben Sie als bewährte Methode für die Verwendung in der Produktion eine Begründung ein, um nachzuverfolgen, warum der Zugriff angefordert wird. Belassen Sie die Standardeinstellungen, und wählen Sie dann „Open Ports“ (Offene Ports), wie in Abbildung 16.6 dargestellt.

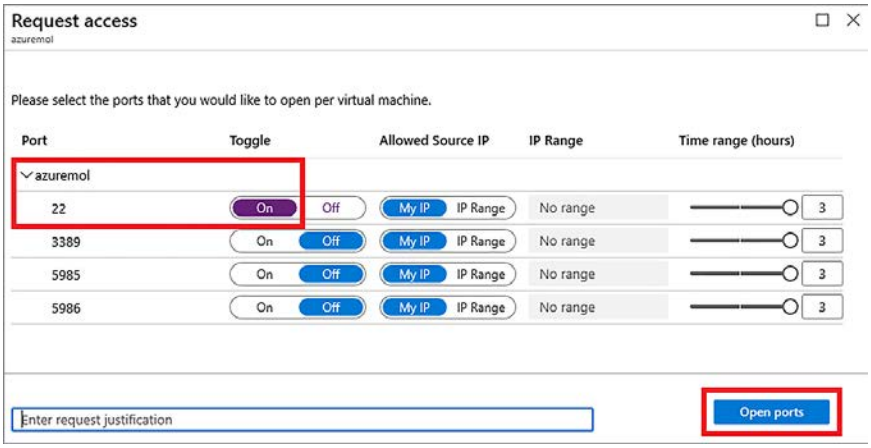


Abbildung 16.6 Wenn Sie JIT aktivieren, können Sie die zulässigen Standardregeln, die zulässigen Quell-IPs und eine maximale Anforderungszeit in Stunden ändern. Diese JIT-Regeln ermöglichen eine detaillierte Kontrolle über das, was erlaubt ist, und ermöglichen nur ein Minimum an Konnektivität.

7 Wenn JIT aktiviert ist, navigieren Sie zu Ihrer Ressourcengruppe, und wählen Sie Ihre VM aus.

8 Wählen Sie „Networking“, um die zugewiesene virtuelle Netzwerkkonfiguration für die VM anzuzeigen. Die Liste der zugeordneten NSG-Regeln wird wie in Bild 16.7 dargestellt.

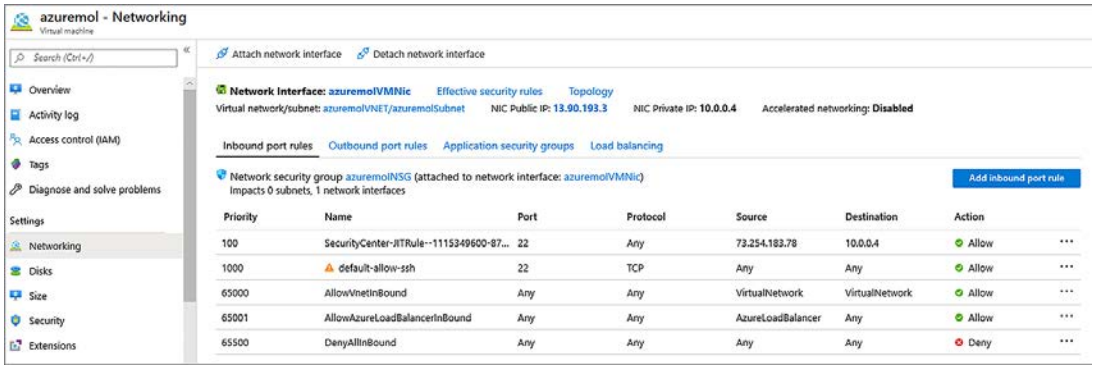


Abbildung 16.7 Die JIT-Regeln werden mit der niedrigsten Priorität angelegt. Diese Prioritäten stellen sicher, dass die JIT-Regeln Vorrang vor allen späteren Regeln auf Subnetzebene haben.

Die JIT-Regeln werden oben in der Liste angezeigt, da sie die niedrigste Priorität haben. Datenverkehr zur IP-Adresse der VM ist zulässig, jedoch nur von Ihrer eigenen IP-Adresse. Das ist das, was JIT konfiguriert hat. Was hier seltsam erscheinen mag, ist, dass eine default-allow-ssh-Regel immer noch existiert und den gesamten Datenverkehr erlaubt. Denken Sie an Kapitel 5 zurück, als wir über NSGs gesprochen haben. Erkennen Sie, was hier passiert?

JIT gilt nur für die VM. In der JIT-Regel zeigt die Zieladresse die IP-Adresse der VM an. In dem in Abbildung 16.7 dargestellten Beispiel ist das 10.0.0.0.0.4. Datenverkehr ist zulässig. Die eigentliche NSG-Regel wird jedoch auf das gesamte Subnetz angewendet. Die default-allow-ssh-Regel gilt für die Subnetzebene und ermöglicht den Datenverkehr von jeder Quelle und jedem Ziel.

NSG-Regeln werden in der Reihenfolge ihrer Priorität verarbeitet, von niedrig bis hoch. Wie in Kapitel 5 besprochen, wird eine deny-Aktion immer wirksam, unabhängig von zusätzlichen Regeln. Selbst wenn Sie diese default-allow-ssh-Regel geändert haben, um den Datenverkehr zu verweigern, würde die JIT-Regel dennoch den Zugriff auf die spezifische VM und die definierte Quell-IP-Adresse zulassen.

Achten Sie auf dieses Layering von NSG-Regeln. Im Idealfall würden Sie die default-allow-ssh-Regel entfernen und dann den Zugriff nur bei Bedarf mit JIT zulassen. Bei diesem Ansatz wird SSH durch die endgültige DenyAllInbound-Regel verweigert. Wenn Sie eine Verbindung mit einer VM herstellen müssen, verwenden Sie JIT, um Zugriff anzufordern. Dieses erstellt automatisch eine Regel, damit SSH für einen definierten Zeitraum auf Ihre IP-Adresse ESS zugreifen kann.

Die NSG-Regel wird nach Ablauf der angegebenen Zeitspanne automatisch gelöscht. Standardmäßig werden die JIT-Regeln für drei Stunden angewendet. Nach Ablauf dieser Zeit kehrt die VM in einen sichereren Zustand zurück, und Sie müssen erneut den Zugriff auf die VM anfordern.

Dieser JIT-Prozess soll steuern, wer den Zugriff auf die VM anfordern und erhalten darf. Aber nur weil eine Person erfolgreich Zugriff auf eine VM anfordern kann, bedeutet das nicht, dass sie die Berechtigung hat, sich bei dieser VM anzumelden. Alles, was in Azure passiert, ist, dass die definierten NSG-Regeln aktualisiert werden. Security Center und JIT können keine Zugangsdaten auf der VM hinzufügen, entfernen oder aktualisieren.

Alle JIT-Anfragen werden ebenfalls protokolliert. Wählen Sie im Security Center die Option „Just in Time VM Access“ (JIT-VM-Zugriff) und dann Ihre Regel. Wählen Sie rechts den Menüpunkt „...“ und dann das Aktivitätsprotokoll. Dieses Aktivitätsprotokoll hilft Ihnen, im Falle eines Problems zu überprüfen, wer den Zugriff auf eine VM beantragt hat.

Der JIT-VM-Zugriff ist eine Möglichkeit, mit dem Security Center und Azure die Sicherheit Ihrer VMs zu gewährleisten. Der Zugriff auf die VMs ist ein wichtiger Bestandteil der Sicherheit. Aber was ist mit den Anwendungen, Bibliotheken und Diensten, die auf den VMs ausgeführt werden? An dieser Stelle müssen Sie sicherstellen, dass alle aktuellen Sicherheitsupdates rechtzeitig auf Ihre VMs angewendet werden.

16.3 Azure Update Management

Ein Bereich, über den das Azure Security Center berichten kann, ist der Status aller erforderlichen Betriebssystemaktualisierungen, die von der VM benötigt werden. Bei Ihrem Pizzageschäft sollten Sie versuchen, die neuesten Sicherheits- und Anwendungspatches zu installieren. Sie wollen schließlich keine Systeme ausführen, die eine bekannte Schwachstelle oder einen bekannten Angriffsbereich aufweisen, sodass eine Möglichkeit, die Aktualisierungen dieser Systeme zu automatisieren und die Einhaltung der Vorschriften zu verfolgen, Ihre Sicherheit steigert. Wenn Sie mit

Anwendungen arbeiten, die Kundendaten und Zahlungsinformationen enthalten, sollten Sie Systeme nicht ohne die neuesten Patches betreiben. Und denken Sie daran, eine Testumgebung zu planen, mit der Sie Sicherheitspatches sicher einspielen und sicherstellen können, dass sie keine Probleme verursachen, bevor Sie sie auf Produktionssysteme anwenden!

Eine Update-Verwaltungsfunktion ist in Azure VMs integriert und kann Betriebssystemaktualisierungen scannen, melden und korrigieren. Das Tolle an dieser Lösung ist, dass sie sowohl unter Windows als auch unter Linux und sogar unter Linux in verschiedenen Distributionen wie Ubuntu, Red Hat und SUSE funktioniert. Abbildung 16.8 zeigt, wie die Update-Verwaltung die erforderlichen Updates überwacht und installieren kann.

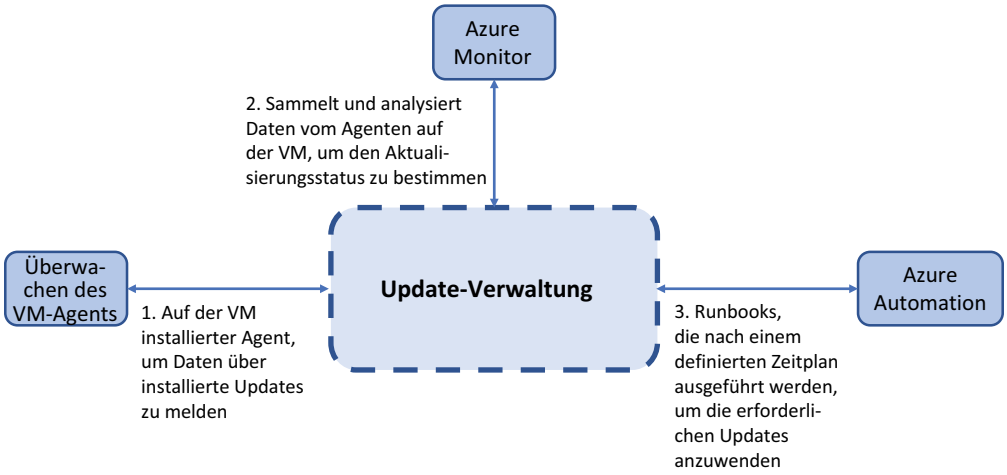


Abbildung 16.8 Die Update-Verwaltung installiert einen VM-Agent, der Informationen über die installierten Updates auf jeder VM sammelt. Diese Daten werden von Azure Monitor analysiert und an die Azure-Plattform zurückgemeldet. Die Liste der erforderlichen Updates kann dann für die automatische Installation über Azure Automation-Runbooks geplant werden.

Es dauert einige Minuten, bis sich die VM vorbereitet hat und über ihren Aktualisierungsstatus Bericht erstattet, also richten wir Ihre VM ein und sehen dann, was hinter den Kulissen vor sich geht.

Jetzt testen

Um Ihre VM für die Update-Verwaltung zu konfigurieren, führen Sie die folgenden Schritte aus:

1. Öffnen Sie das Azure Portal und wählen Sie „Resource Groups“ (Ressourcengruppen) aus dem Menü links.
2. Wählen Sie Ihre Ressourcengruppe aus, z. B. `azuremolchapter16` und wählen Sie dann Ihre VM, z. B. `azuremol`.
3. Wählen Sie unter Vorgänge die Option „Update Management“ (Update-Verwaltung).
4. Akzeptieren Sie die Standardoption für den Ort, und erstellen Sie einen Log-Dateien-Analyse-Arbeitsbereich und ein Automatisierungskonto. Wir untersuchen diese Komponenten im weiteren Verlauf dieses Abschnitts näher.
5. Um die Update-Verwaltung für die VM einzuschalten, wählen Sie „Enable“ (Aktivieren).

Sie gelangen zurück auf die Registerkarte „Update Management Overview“ (Übersicht über das Update Management), aber es dauert einige Minuten, die VM zu konfigurieren und über ihren Status zu berichten. Lesen Sie weiter, und lassen Sie den Prozess laufen.

Beschäftigen wir uns etwas mehr damit, was passieren muss, damit diese Update-Verwaltungslösung funktioniert.

16.3.1 Kombinierte Azure-Verwaltungsdienste

Wenn Sie mit On-Premises Microsoft-Technologien gearbeitet haben, sind Sie vielleicht auf die System Center Suite gestoßen. System Center besteht aus mehreren Komponenten wie Configuration Manager, Operations Manager, Orchestrator und Data Protection Manager. Es gibt noch ein paar andere Bestandteile, aber diese Kernkomponenten bieten Ihnen die Möglichkeit, Folgendes zu tun:

- Definieren von Konfigurationen und Soll-Zustand
- Installieren von Anwendungen und Updates
- Bericht über Gesundheit und Sicherheit
- Automatisierung der Bereitstellung großer Dienste und Anwendungen
- Sichern und Replizieren von Daten

Da Unternehmen in den letzten Jahren zu Cloud Computing-Anbietern gewechselt sind, wurden die traditionelleren On-Premises-System Center-Komponenten von Azure-Diensten ersetzt, damit sie in einer Hybridumgebung betrieben werden können. Wir haben uns bereits in früheren Kapiteln mit zwei Komponenten beschäftigt, auch wenn Sie es nicht bemerkt haben:

- *Azure Backup* bietet eine Möglichkeit, VMs oder einzelne Dateien zu sichern, Aufbewahrungsrichtlinien zu definieren und Daten wiederherzustellen.
- *Azure Site Recovery* ermöglicht es Ihnen, VMs im Falle einer Naturkatastrophe oder eines längeren Ausfalls in verschiedene geografische Regionen zu replizieren.

Sowohl Azure Backup als auch Site Recovery haben Ihnen beim Schutz Ihrer Daten in Kapitel 13 geholfen. Jetzt werden Sie zwei zusätzliche Dienste mit Update Management verwenden:

- *Log Analytics-Arbeitsbereiche* sammeln Informationen von verschiedenen Quellen oder Agents und ermöglichen es Ihnen, Richtlinien und Abfragen zu definieren, um Sie über mögliche Zustände zu informieren. Diese Abfragen und Warnungen können Ihnen helfen, den Aktualisierungsstatus einer VM zu verfolgen oder Sie über Konfigurations- oder Sicherheitsprobleme zu informieren.
- *Azure Monitor* bietet Details und Berichte zu Informationen basierend auf der Verarbeitung in Log Analytics-Arbeitsbereichen. Azure Monitor bietet eine zentrale Möglichkeit zum Anzeigen von Warnungen, zum Abfragen von Protokolldaten und zum Generieren von Benachrichtigungen über alle Azure-Ressourcen hinweg.
- *Azure Automation* ermöglicht es Ihnen, Runbooks zu erstellen, die Befehle oder ganze Skripte ausführen. Runbooks können große, komplexe Implementierungen sein und mehrere andere Runbooks aufrufen. Wir gehen in Kapitel 18 ausführlich auf Azure Automation ein.

Die Integration dieser Komponenten ist in Abbildung 16.9 dargestellt.

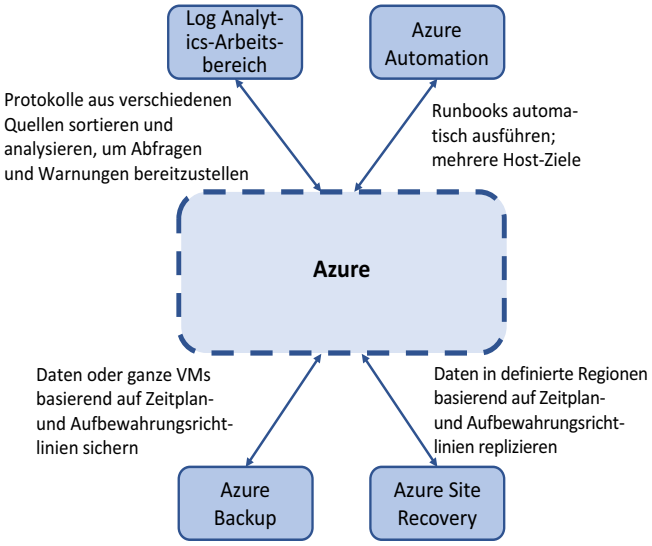


Abbildung 16.9 Mehrere Azure-Dienste arbeiten zusammen, um Management- und Konfigurationsfunktionen für Ihre gesamte Anwendungsumgebung bereitzustellen. Die Dienste, die diese Komponenten verwenden, sind nicht auf Azure VMs oder Ressourcen beschränkt und können bei entsprechender Konfiguration über andere Cloudanbieter oder On-Premises Systeme hinweg funktionieren.

Sowohl die Log-Datei-Arbeitsbereiche als auch Azure Automation sind leistungsstarke Komponenten und können problemlos ganze Kapitel eines Buchs selbst füllen. Da nur eine Handvoll VMs zu verwalten sind, übersieht man leicht, dass ein zentrales Log-Repository für Abfragen und Warnungen oder eine Möglichkeit zur Automatisierung von Konfigurationen und Implementierungen über VMs hinweg erforderlich ist. Wenn Sie noch keine Liste der Azure-Komponenten erstellt haben, die Sie weiterverfolgen können, wenn Sie mit diesem Buch fertig sind, starten Sie eine und fügen Sie diese beiden Komponenten zu dieser Liste hinzu!

Wichtig zu verstehen ist es außerdem, dass es in Azure mehrere Dienste und Komponenten gibt, die interagieren und sich gegenseitig ergänzen können. So wie Azure VMs und Azure virtuelle Netzwerke einzelne Dienste sind, ergänzen oder verlassen sich beide Dienste auch aufeinander. Azure Backup und die Azure-Diagnoseerweiterung sind großartige Einzelkomponenten, aber sie glänzen erst richtig, wenn die Log Analytics-Workspaces und verwendet Azure Monitor werden, um ihren Status zu überwachen und alle erzeugten Ereignisse oder Warnungen zu sammeln. Hoffentlich haben Sie begonnen, einige dieser verwandten Komponenten zu identifizieren und zu sehen, wie Azure-Dienste oft aufeinander aufbauen. Nun, da wir uns mit diesen letzten Kapiteln befassen und bei den Sicherheits- und Überwachungsoptionen sind, ist es unser Ziel, sicherzustellen, dass die Anwendungen, die Sie in Azure ausführen, gesund und stabil sind.

Diese kleine Sache namens „Identität“

Wenn man an Dienstleistungen denkt, die sich gegenseitig ergänzen, ist ein großer (und ich meine, wirklich *groß*!) Teil von Azure, den wir kurz angeschnitten haben, das Azure Active Directory (Azure AD). Identität ist für alles in Azure von zentraler Bedeutung, und AAD bietet einige der Sicherheitsfunktionen, die wir in Kapitel 6 mit dem Azure Resource Manager-Bereitstellungsmodell untersucht haben. Die Möglichkeit, mit RBACs zu begrenzen, welche Aktionen eine Ressource von bestimmten Benutzern oder Gruppen durchgeführt werden kann, ist in eine zentrale Identitätslösung eingebunden. Selbst die Möglichkeit, sich beim Azure Portal oder der Azure CLI anzumelden, wird von Azure AD gesteuert.

Dieses Buch behandelt Azure AD nicht, da der Umfang des Angebots breit gefächert und ganz anders ist als bei Azure IaaS- und PaaS-Diensten wie VMs, Scale Sets und Web Apps. Es mag einige Überschneidungen bei den Lesern dieser Themen geben, aber die meisten Entwickler haben ein anderes Ziel für das, wenn es um Azure AD geht, als ein Anwendungsmanager oder ein IT-Profi, der die Infrastruktur bereitstellt.

Je nach Azure-Konto können Ihre Anwendungsmöglichkeiten von Azure AD eingeschränkt sein. Wenn Sie sich für ein kostenloses Azure-Testkonto anmelden, wird eine Standard-Azure AD-Instanz für Sie erstellt. Sie sind das primäre Konto in diesem Verzeichnis und haben volle Administrationsrechte. Wenn Sie sich mit einem Konto Ihres Unternehmens oder Ihrer Bildungseinrichtung bei Azure anmelden, besteht eine gute Chance, dass Sie nur wenige bis gar keine Administratorrechte haben. Selbst wenn wir uns also auf ein paar Themen einigen könnten, auf die Sie sich konzentrieren sollten, können Sie eventuell keine der Übungen direkt durchführen. Und ich empfehle Ihnen wirklich nicht, in einer echten Azure AD-Umgebung herumzustochern, um zu lernen, wie alles funktioniert!

Aber Azure AD ist ein weiterer dieser zentralen Dienste in Azure, der viele andere Dienste und Komponenten miteinander verbindet. Cloud Computing macht die Dinge nicht auf magische Weise einfacher oder zerlegt betriebliche Silos – Sie benötigen immer noch die Fähigkeiten, um mit verschiedenen Teams und Interessengruppen zusammenzuarbeiten. In diesen Kapiteln haben Sie hoffentlich die Kernkompetenzen für diese Azure-Dienste erworben, die Ihnen helfen werden, zu verstehen, wie Sie große, redundante Anwendungen erstellen und auf einem besseren Niveau und mit mehr Bewusstsein dafür, was andere Teams zu bewältigen haben, kommunizieren können.

16.3.2 Überprüfen und Anwenden von Updates

Es kann einige Zeit in Anspruch nehmen, bis der VM-Agent den ersten Scan durchgeführt hat und über den Status der angewandten Updates berichtet. Die Liste der installierten Komponenten muss auch mit der Liste der verfügbaren Updates für ein bestimmtes Betriebssystem und eine bestimmte Version verglichen werden. Wenn Ihre VM noch nicht fertig ist und über ihren Status berichtet hat, lesen Sie weiter, und schauen Sie in ein paar Minuten wieder vorbei. Wenn sie fertig ist, sieht die Übersicht wie in Abbildung 16.10 aus. Seien Sie geduldig, es kann 10 bis 15 Minuten dauern, bis die Agent-Bereitschaft als „Ready“ (Bereit) angezeigt wird, und Sie können Updates für die Installation planen.

Eine Liste der erforderlichen Updates ist großartig, aber wie sieht es mit der Möglichkeit aus, sie zu installieren? Hier setzt Azure Automation an! Wenn Sie die Update-Verwaltung aktiviert haben, werden eine Reihe von Azure Automation-Runbooks erstellt, die den Prozess zum Übernehmen der erforderlichen Updates automatisch abwickeln.

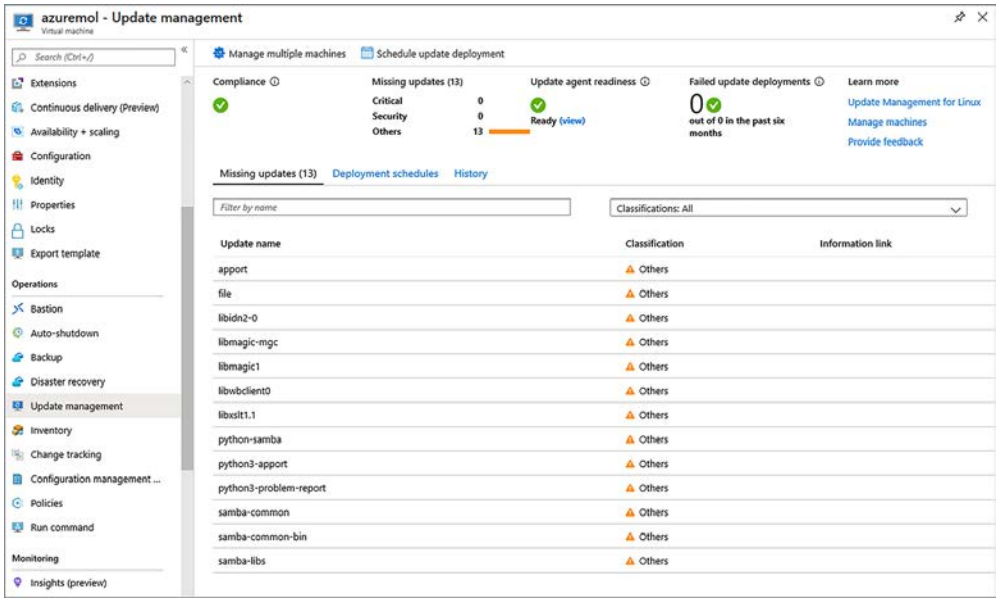


Abbildung 16.10 Sobald der VM-Agent die Konformität überprüft hat, wird eine Liste der verfügbaren Updates bereitgestellt. Je nach Betriebssystem und Version kann die Update-Verwaltung mit dem Log Analytics-Arbeitsbereich und Azure Monitor zusammenarbeiten, um die Updates nach Schweregrad zu klassifizieren oder Links zu den entsprechenden Update-Hotfixseiten bereitzustellen.

Jetzt testen

Wenn Sie Glück haben (oder Pech), kann Ihre VM melden, dass keine Updates erforderlich sind. VM-Images werden in Azure häufig aktualisiert, und wenn Sie eine VM kurz nach dem Erstellen des letzten Images bereitstellen, sind alle erforderlichen Updates bereits installiert. Wenn dem so ist, dann lesen Sie diese Schritte durch, damit Sie verstehen, was erforderlich ist, wenn Ihre VMs aktualisiert werden müssen!

Um die erforderlichen Updates für Ihre VM zu installieren, führen Sie die folgenden Schritte aus:

1. Wählen Sie im Abschnitt „Update Management“ (Update-Verwaltung) Ihrer VM die Option „Schedule Update Deployment“ (Update-Bereitstellung planen).
2. Geben Sie einen Namen für die Update-Bereitstellung ein, z. B. `azuremolupdates`, und überprüfen Sie dann die Update-Klassifizierungen. Sie können steuern, welche Updates angewendet werden. Lassen Sie vorerst alle voreingestellten Optionen unverändert.
3. Mit „Updates to Exclude“ (Auszuschließende Updates) können Sie bestimmte Updates angeben, die Sie nicht installieren möchten. Wenn Sie wissen, dass Ihre Anwendung eine bestimmte Version eines Pakets oder einer Bibliothek benötigt, können Sie sicherstellen, dass kein aktualisiertes Paket installiert ist, das Probleme verursacht. Überprüfen Sie die verfügbaren Optionen, in dieser Übung gibt es aber nichts zu ändern.

4 Wählen Sie „Schedule Settings“ (Zeitplaneinstellungen), und wählen Sie dann aus den Kalender- und Zeitoptionen einen Zeitpunkt für die Anwendung der Updates. Die Startzeit muss mindestens 5 Minuten vor der aktuellen Zeit liegen, um der Azure-Plattform einige Augenblicke für die Verarbeitung und Planung Ihres Runbooks in Azure Automation zu geben.

5 Wenn Sie bereit sind, wählen Sie OK.

6 Wenn bestimmte Anwendungen und Dienste angehalten oder heruntergefahren werden müssen, bevor Updates angewendet werden, und erneut gestartet werden müssen, wenn die Updates abgeschlossen sind, wählen Sie „Pre-scripts + Post-scripts“ (Pre-Skripts + Post-Skripts). Separate Automatisierungsaufgaben können so konfiguriert werden, dass Aktionen auf den VMs vor und nach der Anwendung der Updates durchgeführt werden.

7 Das Wartungsfenster (Minuten) definiert, wie lange der Aktualisierungsprozess laufen kann, bevor die VM wieder in Betrieb genommen werden muss. Dieses Wartungsfenster verhindert lang laufende Aktualisierungsprozesse, die dazu führen können, dass eine VM stundenlang nicht verfügbar ist. Sie können das Wartungsfenster kürzer oder länger machen, abhängig von Service Level Agreements für die Anwendungen, die auf diesen VMs ausgeführt werden, oder der Anzahl und Größe der erforderlichen Updates. Übernehmen Sie den Standardwert, und wählen Sie dann „Create“ (Erstellen) aus.

8 Wählen Sie im Fenster „Update Management“ (Update-Verwaltung) die Option „Deployment Schedules“ (Bereitstellungszeitpläne). Die Updates werden wie geplant für die Installation an dem von Ihnen gewählten Datum und der Uhrzeit aufgelistet, wie in Abbildung 10.86.11 dargestellt.

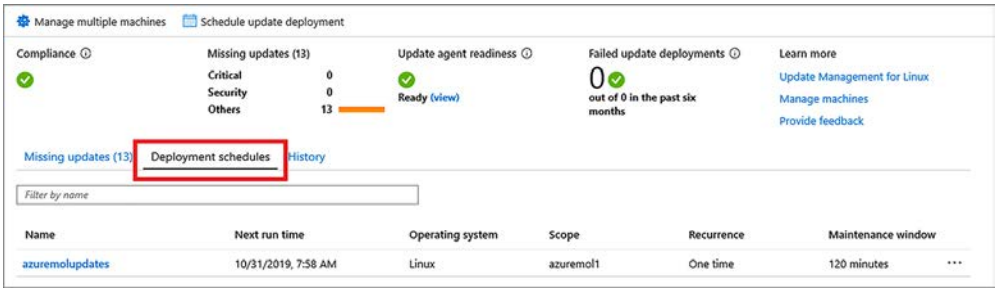


Abbildung 16.11 Die Liste der geplanten Bereitstellungsaufgaben wird angezeigt. Auf Wunsch können Sie eine bestimmte Aufgabe löschen, ansonsten werden die Updates automatisch zum definierten Zeitpunkt übernommen.

9 Wählen Sie oben im Fenster Update-Verwaltung die Option „Manage Multiple Machines“ (Mehrere Maschinen verwalten) aus. Das Fenster wechselt zum Azure Automation-Konto, das erstellt wurde, wenn die Updateverwaltung für die VM aktiviert wurde. Machen Sie sich im Moment keine allzu großen Sorgen über die Funktionen der Runbooks. Es gibt nichts, was Sie anpassen können, und wir sehen uns Azure Automation in Kapitel 18 an.

Beachten Sie, dass Sie „Add Azure VM“ (Azure-VM hinzufügen) oder „Add Non-Azure machine“ (Nicht-Azure-Maschine hinzufügen) wählen können, wie in Abbildung 16.12 dargestellt. Diese Möglichkeit zeigt einen einzigen Ansatz für die Verwaltung von Updates in Ihrer gesamten Anwendungsumgebung, nicht nur für Azure VMs.

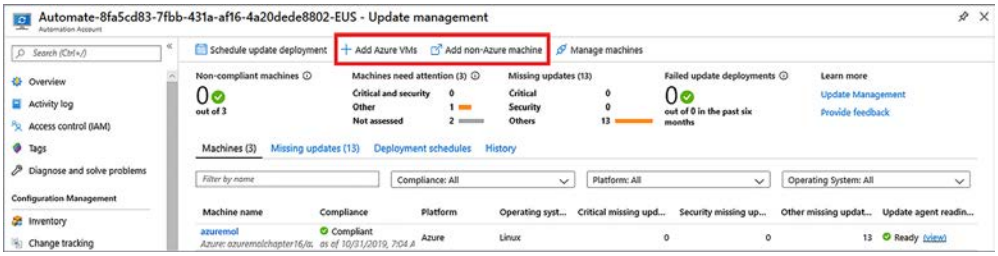


Abbildung 16.12 **Im Azure Automation-Konto können Sie mehrere Maschinen verwalten, sich ihren Status anzeigen lassen oder Updates anwenden. Sowohl Azure VMs als auch Nicht-Azure-Maschinen können von demselben Azure Automation-Konto überwacht und gesteuert werden. Hinter den Kulissen kann Azure mit anderen Anbietern zusammenarbeiten, um Agents auf Maschinen in einer Hybrid-Umgebung zu installieren. Diese Integration ermöglicht es, dass eine einzige Dashboard- und Verwaltungsplattform für Ihre Aktualisierungsanforderungen zur Verfügung steht.**

10 Gehen Sie zurück zum Fenster „Update Management“ (Update-Verwaltung) für Ihre VM, und wählen Sie die Registerkarte „History“ (Verlauf). Wenn Ihre Update-Bereitstellung beginnt, wird der Status angezeigt. Denken Sie daran, dass Sie den Auftrag so geplant haben, dass er einige Minuten in der Zukunft ausgeführt wird, sodass er nicht sofort angezeigt wird.

11 Wählen Sie den Zeitplan aus, um den Status und die Ausgabe anzuzeigen, wie in Abbildung 16.13 dargestellt.

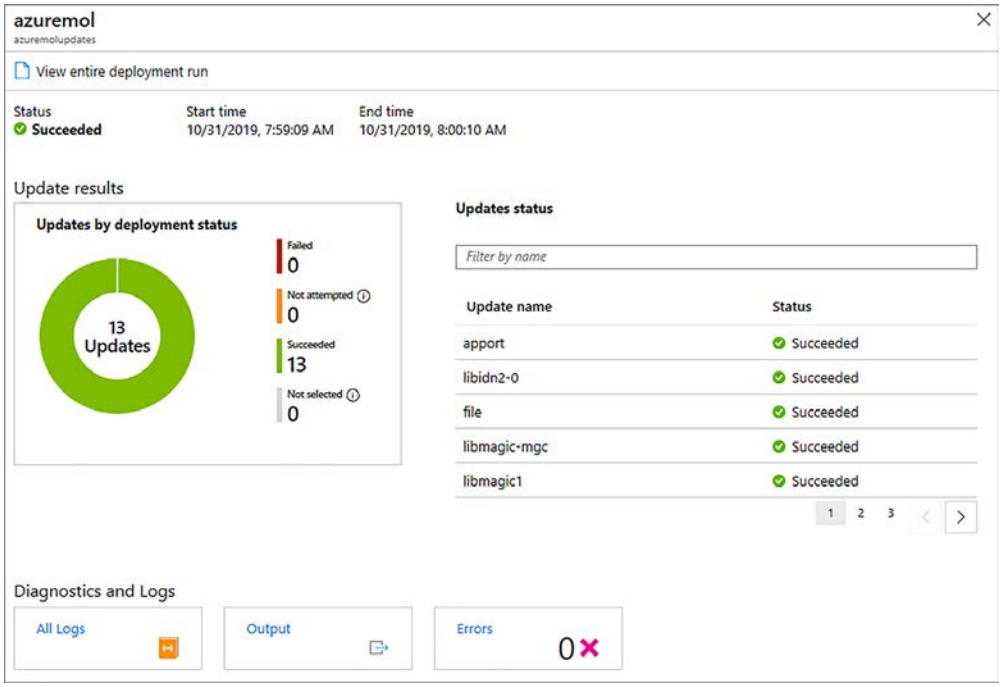


Abbildung 16.13 **Sie können den Status der ausgeführten Azure Automation-Aufträge im Portal überwachen. Um Aufgaben zu überprüfen oder Fehler zu beheben, können Sie auf einen Auftrag klicken, um alle Ausgaben und generierten Log-Dateien anzuzeigen.**

12 Wenn die Update-Bereitstellung abgeschlossen ist, navigieren Sie zurück zu Ihrer Ressourcengruppe, wählen Ihre VM aus und dann „Update Management“ (Update-Verwaltung). Es kann einige Minuten dauern, bis sich der Agent selbst aktualisiert und über den Log-Datei-Arbeitsbereich meldet, dass die Updates angewendet wurden. Dann sollte er im Dashboard anzeigen, dass die VM auf dem neuesten Stand ist und keine zusätzlichen Updates erforderlich sind.

Dieses Kapitel war eine wilde Tour durch das Security Center und die damit verbundenen Komponenten wie JIT-VM-Zugriff und Updateverwaltung. Das Ziel ist, dass Sie anfangen, über die Bereitstellung und Ausführung einer VM oder Webanwendung hinaus zu denken und stattdessen die damit verbundene breitere Anwendungsverwaltung zu planen. Cloud Computing ändert nichts an der Notwendigkeit von Sicherheitsrichtlinien. Es besteht unumstritten ein größerer Bedarf an der Sicherung von Ressourcen. Lassen Sie sich von Azure-Funktionen wie dem Security Center durch die zu erledigenden Aufgaben führen, und verwenden Sie die integrierten Tools wie Update-Verwaltung und Azure Automation, um jederzeit die Sicherheit zu gewährleisten.

16.4 Übung: Aktivieren von JIT und Updates für eine Windows-VM

Dieses Kapitel behandelte einige Komponenten, die einige Zeit in Anspruch genommen haben können, um sich selbst zu aktivieren und über ihren erwarteten Status zu berichten. Dieses Übung ist optional und soll mehr zeigen, dass es nichts Betriebssystemspezifisches an diesen Funktionen gibt. Wenn Sie keine Zeit haben oder das Gefühl haben, dass Sie verstehen, wie Sie diese Funktionen auch auf eine Windows-VM anwenden können, können Sie diese Übung gerne überspringen. Andernfalls sollten Sie die folgenden Aufgaben ausprobieren, um zusätzliche Übungen mit dem Security Center und der Update-Verwaltung zu erhalten. Übung macht den Meister, oder?

1 Erstellen Sie eine Windows Server-VM Ihrer Wahl in der gleichen Ressourcengruppe, die Sie für die vorherigen Übungen verwendet haben, wie z. B. `azuremolchapter16`.
2 Zeigen Sie die NSG-Regeln für das VM/Subnetz an, und löschen Sie alle Standardregeln, die RDP auf TCP-Port 3389 zulassen.
3 Verwenden Sie dann Ihren lokalen Remote Desktop Client, um sicherzustellen, dass RDP-Verbindungen blockiert sind.
4 Fordern Sie JIT-Zugriff an, überprüfen Sie die NSG-Regeln erneut, und bestätigen Sie, dass Sie nun RDP auf Ihrer VM ausführen können.
5 Aktivieren Sie die Update-Verwaltung auf Ihrer Windows-VM. Diesmal sollten Sie in der Lage sein, den vorhandenen Log Analytics-Arbeitsbereich und die Konten von Azure Automation zu verwenden.
6 Lassen Sie den Überwachungs-Agent über erforderliche Aktualisierungen berichten, und planen Sie dann die Aktualisierungen, die über Azure Automation ausgeführt werden sollen.

Teil 4:

Das coole Zeug

Das ist wirklich eine tolle Sache! In diesen letzten Kapiteln lernen wir einige der kommenden Technologien kennen, die Sie in Azure einsetzen können, wie künstliche Intelligenz und Machine Learning, Container und Kubernetes sowie das Internet der Dinge. Diese Dienste nutzen Sie oder Ihr Unternehmen vielleicht gerade nicht, aber mit den aktuellen Trends beim Computing werden sie es wahrscheinlich bald tun. Diese Dienste sind einige der aufregendsten Technologien, mit denen Sie derzeit arbeiten können. Obwohl dieses Buch versucht, diese Themen in der Mittagspause abzudecken, ist dieser Teil gut geeignet, um die Dinge zusammenzufassen und Ihnen die Möglichkeiten zu zeigen, was Sie in Azure entwickeln können.

17 Machine Learning und künstliche Intelligenz

Hoffentlich werden wir nicht in einer Welt landen, in der Filme wie *Terminator* und *Matrix* wahr werden. In diesen Filmen verursacht der Aufstieg der künstlichen Intelligenz (KI) fast den Untergang der Menschheit, da Maschinen darum kämpfen, die Kontrolle über ihre Umgebung zu übernehmen. Ein Grund zur Sorge beim Computing ist derzeit, wie die Entwicklung der KI meist von großen, privaten Unternehmen durchgeführt wird, mit wenig oder gar keiner Regulierung oder zentralen Überwachung. Das soll aber natürlich nicht heißen, dass KI etwas schlechtes ist! Digitale Assistenten auf Smartphones können bei vielen alltäglichen Aufgaben helfen. Machine Learning (ML) in Navigationsanwendungen kann die tägliche Fahrt der Benutzer*innen überwachen, um alternative Routen basierend auf Straßen- oder Wetterbedingungen vorzuschlagen. Durch Steuerung kann die Heizung automatisch basierend auf der Außentemperatur, Tageszeit und Jahreszeit (z. B. Sommer oder Winter) angepasst werden.

Wenn Sie diesen letzten Teil des Buchs beginnen, werden Sie mehr über die Azure-Dienste für Machine Learning und künstliche Intelligenz erfahren. In einem Kapitel. In Ihrer Mittagspause. Lassen Sie uns einige realistische Erwartungen setzen: Sie werden in den nächsten 45 Minuten kein Experte für ML oder KI werden! Sie können, wenn Sie Ihr Sandwich schnell essen, genug über die vielen Dienste erfahren, die Azure anbietet, um zu verstehen, wie Sie einige dieser ML- und KI-Dienste in Ihre Anwendungen integrieren können. Viele der Azure ML- und KI-Dienste benötigen zumindest einige Vorkenntnisse bei Datenalgorithmen, Programmiersprachen, Batch-Verarbeitung oder Sprachverständnis, also rechnen Sie nicht damit, in der nächsten Stunde zum Profi zu werden!

In diesem Kapitel begeben wir uns auf eine turbulente Tour durch einige der Cognitive Services von Azure, die ML- und KI-Funktionen bieten. Sie erfahren, wie Sie diese Dienste verwenden, um grundlegendes Machine Learning an Datenmodellen durchzuführen, und dann verwenden Sie ein wenig vom Azure Web Apps-Dienst und dem Microsoft Bot Framework, um einige der KI-Dienste anzuwenden, die einen Pizza-Shop-Bot für Kunden ausführen können, um Pizza zu bestellen.

17.1 Überblick zu und Beziehung von KI und ML

Halten Sie sich fest, denn wir sind dabei, von 0 auf 600 km/h in nur wenigen Seiten zu beschleunigen! KI und ML überschneiden sich oft, wenn Sie Anwendungen in Azure erstellen. Lassen Sie uns erforschen, was jede davon ist, und uns dann Gedanken darüber machen, wie sie zusammenarbeiten.

17.1.1 Künstliche Intelligenz

KI ermöglicht es Computern, Aufgaben mit einem gewissen Maß an Flexibilität und Bewusstsein zu erledigen und ihre Entscheidungen basierend auf externen Faktoren oder ohne die Notwendigkeit menschlicher Interaktion anzupassen. Das Ziel ist in der Regel nicht der Aufbau eines völlig autonomen Systems, das sich entwickeln und Gedanken für sich selbst entwickeln kann, sondern die Verwendung einer Reihe von Datenmodellen und Algorithmen, die den Entscheidungsprozess steuern.

Zu den gängigen KI auf PCs und Smartphones gehören Siri, Cortana und Google Assistant. Wie in Abbildung 17.1 dargestellt, ermöglichen diese KI-Ressourcen die Kommunikation, oft per Sprachbefehl, um nach Anweisungen zu fragen, Erinnerungen zu setzen, das Web zu durchsuchen und vieles mehr.

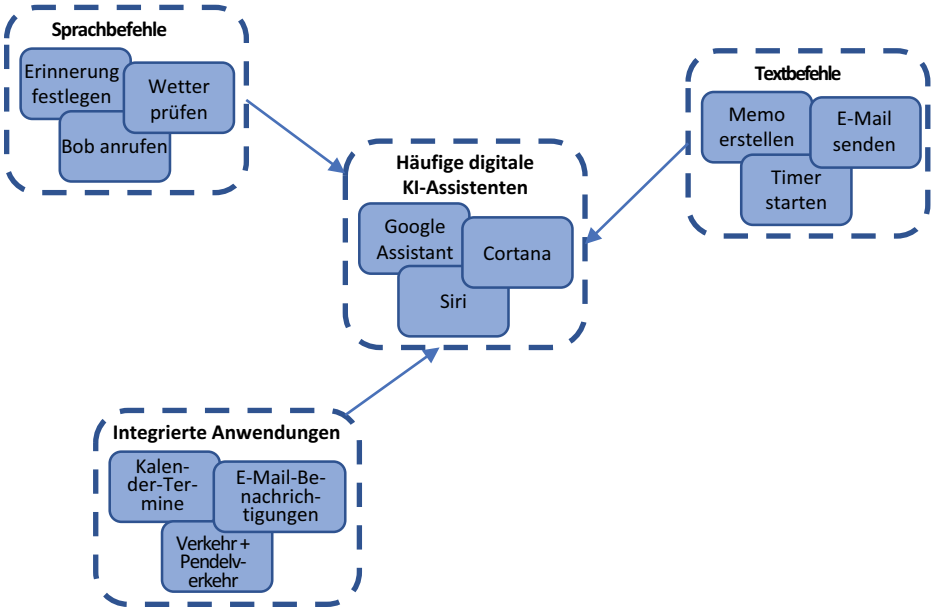


Abbildung 17.1 Eine gängige Anwendung der KI im Alltag sind digitale Assistenten wie Cortana, Siri und Google Assistant. Sie können Sprach- oder Textbefehle verwenden, um mit ihnen zu interagieren, und sie können Ihren Tageskalender oder die Verkehrslage überwachen, um Sie vor Stau zu warnen.

Digitale Assistenten wie diese beinhalten in der Regel nicht viel von dem, was Sie vielleicht für *Intelligenz* halten. Sie hören zu und reagieren auf Ihre Eingaben. Diese Eingaben können jedoch variieren und sind nicht immer spezifische Befehle. Denken Sie daran, wie Sie mit einem digitalen Assistenten eine Erinnerung einrichten. Sie können einen der folgenden Sätze benutzen:

- „Erinnere mich um 5 Uhr daran, Milch zu holen."
- „Sag mir, ich soll auf dem Heimweg Milch besorgen."
- „Ich muss Milch kaufen, wenn ich im Laden bin."

Wenn Sie eine traditionelle Anwendung entwickelt haben, müssen Sie Code schreiben, der alle möglichen Variationen der Art und Weise abdeckt, wie ein Benutzender Anweisungen geben kann. Sie könnten reguläre Ausdrücke erstellen, um einige der Variationen zu erfassen, aber was passiert, wenn der Benutzende sich einen Satz ausdenkt, den Sie nicht programmiert haben? Oder was ist, wenn er den Text eingibt und einen Tippfehler in der Anfrage hat, den Sie nicht erwartet haben? Diese Art von Interaktionen eignen sich hervorragend für die KI. Wie in Abbildung 17.2 dargestellt, ist die Anwendung für mehrere gängige Sätze programmiert und kann dann eine fundierte Einschätzung vornehmen, basierend auf dem, was sie „denkt", was der Benutzende verlangt.

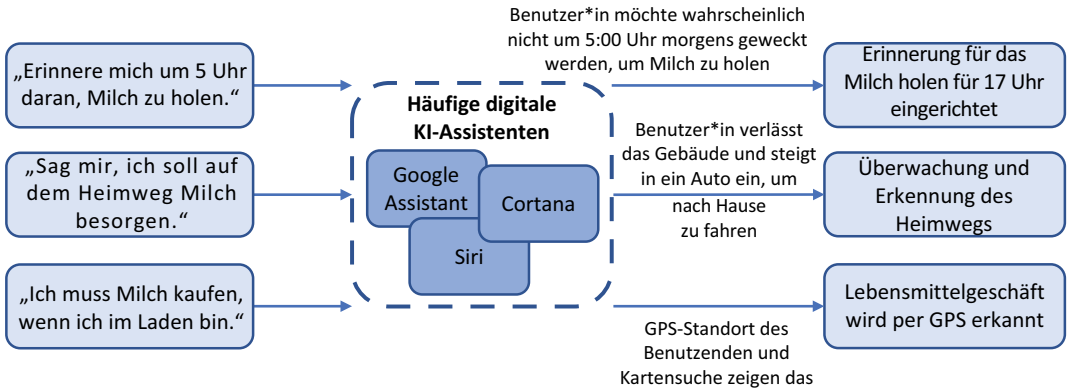


Abbildung 17.2 Die KI erhält eine Eingabe vom Benutzenden und kann Entscheidungen treffen, die am besten zu den erwarteten Maßnahmen passen. Die KI ist nicht mit all diesen möglichen Antworten und Entscheidungsbäumen vorprogrammiert. Stattdessen verwendet sie Datenmodelle und Algorithmen, um den Kontext auf die Benutzungseingaben anzuwenden und die Bedeutung und das passende Ergebnis zu interpretieren.

Es ist (noch) keine wahre Intelligenz, auch nicht in komplexen Formen der KI, sondern eine fundierte Vermutung auf der Grundlage eines Datenmodells, mit dem die KI trainiert wurde. Dieses Datenmodell kann viele Variationen und Sätze beinhalten und im Laufe der Zeit neue Bedeutungen lernen. Wie lernt sie und woher kommen diese Datenmodelle? Hier wird ML wichtig.

17.1.2 Machine Learning

Ein großes Schlagwort im Computing-Bereich in den letzten Jahren war *Big Data*. Das Konzept dabei bezieht sich darauf, dass Computersysteme, insbesondere in der Cloud, eine große Ressource für die Verarbeitung großer Datenmengen darstellen. *Wirklich* große Datenmengen. Diese Verarbeitungsaufträge können je nach Datenmenge und Berechnungsaufwand einige Minuten oder Stunden dauern und ermöglichen es Ihnen, große Datenmengen vorzubereiten und zu analysieren, um bestimmte Muster und Zusammenhänge zu erkennen. Diese Erkenntnisse bilden Datenmodelle, die

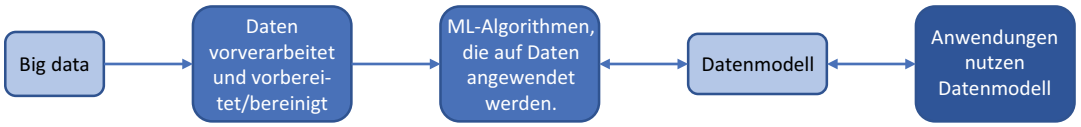


Abbildung 17.3 Große Mengen an Rohdaten werden verarbeitet und einsatzbereit gemacht. Je nach Rohdaten können verschiedene Aufbereitungstechniken und Datenbereinigungen angewendet werden. ML-Algorithmen werden dann auf die vorbereiteten Daten angewendet, um ein geeignetes Datenmodell zu erstellen, das die beste Korrelation zwischen allen Datenpunkten widerspiegelt. Verschiedene Datenmodelle können im Laufe der Zeit erstellt und verbessert werden. Anwendungen können die Datenmodelle auf ihren eigenen Dateneingaben verwenden, um ihre Entscheidungen zu steuern und Muster zu verstehen.

Anwendungen oder die KI als Entscheidungshilfe nutzen können. Wie in Abbildung 17.3 dargestellt, beinhaltet ML einige Schritte und sowohl Input als auch Output.

So funktioniert die grundlegendste Form von ML:

1. Zu Beginn des Prozesses werden große Mengen an Rohdaten als Eingabe bereitgestellt.
2. Diese Daten werden verarbeitet und in ein verwendbares Format gebracht, um sich auf die für die Analyse erforderlichen spezifischen Datenpunkte zu konzentrieren.
3. ML-Algorithmen werden auf die Daten angewendet. An dieser Stelle geht es mit der echten Zahlenberechnung los. Die Algorithmen wurden entwickelt, um Ähnlichkeiten oder Unterschiede über die große Anzahl von Datenpunkten zu erkennen und zu berechnen.
4. Basierend auf der Analyse der Algorithmen wird ein Datenmodell erstellt, das Muster innerhalb der Daten definiert. Diese Datenmodelle können im Laufe der Zeit verfeinert werden, wenn sich Teile des Modells als falsch oder unvollständig erweisen, wenn zusätzliche reale Daten verwendet werden.
5. Anwendungen verwenden die Datenmodelle, um ihre eigenen Datensätze zu verarbeiten. Diese Datensätze sind in der Regel viel kleiner als die Rohdaten, die den ML-Algorithmen zur Verfügung gestellt werden. Wenn das Datenmodell gültig ist, kann bereits mit einer kleinen Dateneingabe aus der Anwendung das richtige Ergebnis oder die richtige Korrelation ermittelt werden.

ML beinhaltet oft komplexe Algorithmen, die darauf ausgelegt sind, alle bereitgestellten Datenpunkte zu verarbeiten. Hadoop und Apache Spark sind zwei gängige Anwendungs-Stacks, die zur Verarbeitung großer Datenmengen verwendet werden. Azure HDInsight ist ein verwalteter Dienst, mit dem Sie die großen Datensätze analysieren können, die von diesen Application Stacks verarbeitet werden. Um die Analyse und die Algorithmen etwas zu vertiefen, ist die Programmiersprache R bei Datenwissenschaftler*innen üblich. Sie hilft bei der Entwicklung der erforderlichen Modelle. Machen Sie sich keine allzu großen Sorgen darüber, was Hadoop oder R sind. Der wichtigste Punkt ist, dass Azure die in der Branche gängigen ML-Tools ausführen kann.

17.1.3 Zusammenbringen von KI und ML

Eine gängige Anwendung auf einem Smartphone ist die Navigations-App, wie in Abbildung 17.4 dargestellt. Ihr Anbieter, wie z. B. Google, kann die Route, die Sie jeden Tag zur Arbeit nehmen, wann Sie normalerweise Ihr Zuhause verlassen und wie lange es dauert, bis Sie dort ankommen, nachverfolgen.

Dieses Google Maps-Beispiel zeigt die Zusammenarbeit von KI und ML. Die KI wird angewendet, um zu herauszufinden, wann eine Benachrichtigung basierend auf den nach der Verarbeitung des ML-Datenmodells

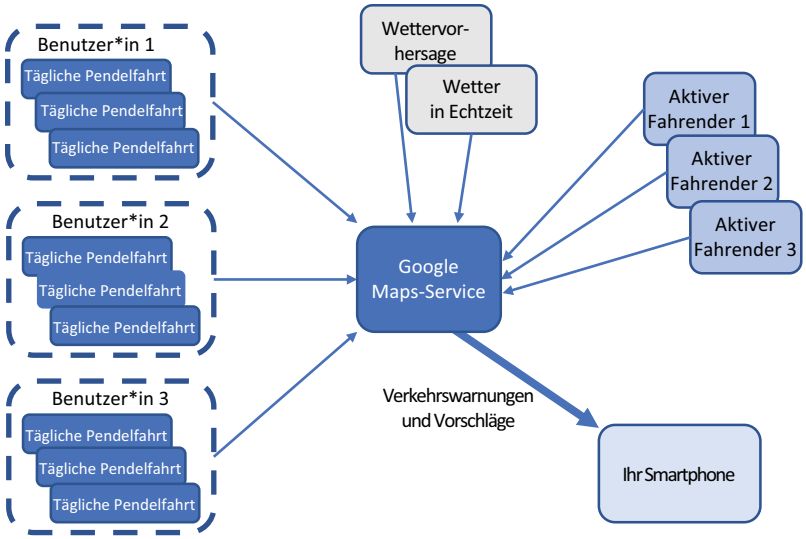


Abbildung 17.4 Der Google Maps-Dienst erhält jeden Tag mehrere Datenpunkte von Nutzer*innen, die Details ihrer Fahrt aufzeichnen. Diese Daten können aufbereitet und verarbeitet werden, ebenso wie die Wettervorhersage und das Wetter in Echtzeit während dieser Fahrten. ML-Algorithmen können auf diese großen Datensätze angewendet werden und ein Datenmodell erstellen. Wenn eine kleinere Stichprobe aktiver Autofahrer ihre aktuellen Reisebedingungen oder Wetterdaten in den Google Maps-Dienst einspeist, kann das Datenmodell zur Vorhersage der Fahrtstrecke verwendet werden und eine Verkehrswarnung auf dem Smartphone generieren, die eine alternative Route nach Hause vorschlägt.

empfangenen Daten zu erstellen ist. Ein weiteres Beispiel für die Zusammenarbeit von KI und ML ist unser vorheriges Beispiel – die Erinnerung an den Kauf von Milch. Wenn die KI mit ML-Datenmodellen trainiert wurde, würde der Assistent wissen, dass Sie wahrscheinlich Milch im Lebensmittelgeschäft kaufen, sodass er Sie nicht daran erinnern würde, wenn Sie zum Baumarkt gingen. Das ML-Datenmodell würde auch der KI helfen zu verstehen, dass es eine sehr große Wahrscheinlichkeit gibt, dass Sie um 17:00 Uhr an etwas erinnert werden wollen, nicht um 5:00 Uhr, also sollte der Assistent Sie nicht um 5:00 Uhr morgens zum Milchkauf wecken. Wenn Ihr Smartphone verfolgt, dass Sie um 17:00 Uhr in Ihr Auto steigen und anfangen, von der Arbeit wegzufahren, generiert ML ein Datenmodell, das voraussagt, dass Sie nach Hause fahren. Das wäre also ein guter Zeitpunkt für die KI, Sie an den Milchkauf zu erinnern.

Dies sind grundlegende, aber wirkungsvolle Beispiele, die zeigen, wie ML zur Verbesserung der KI eingesetzt wird. Sie trainieren die KI, indem Sie einen Satz von Datenpunkten bereitstellen, die von ML verarbeitet werden, um die Genauigkeit oder Entscheidungen zu verbessern.

17.1.4 Azure ML-Tools für Datenwissenschaftler*innen

Ich möchte kurz ein paar Möglichkeiten aufzeigen, wie man in der Praxis mit Zahlen und ML arbeiten kann. Um dieses Kapitel für alle zugänglich zu machen, verwenden die Übungen das Microsoft Bot Framework for AI und ML mit Language Understanding Intelligent Service (LUIS). Damit Sie Ihre ersten Schritte mit ML gehen können, müssen wir uns etwas mehr auf die Datenverarbeitung und Algorithmen konzentrieren.

In Azure gibt es eine Reihe von coolen Komponenten, die Ihnen helfen, im großen Maße in die Daten einzudringen. Erstens gibt es Azure Machine Learning selbst, einen webbasierten Dienst, mit dem Sie Experimente visuell aufbauen können, indem Sie Datensätze und Analysemodelle hinzufügen. Diese Experimente können Datenquellen wie Hadoop und SQL nutzen. Und es gibt zusätzliche Programmierunterstützung durch Sprachen wie R und Python. Sie können Datenquellen, Datenaufbereitungstechniken und ML-Algorithmen per Drag & Drop verschieben. Sie können diese Algorithmen anpassen und dann die erzeugten Datenmodelle überprüfen und optimieren.

Azure Machine Learning bietet eine niedrige Barriere für den Zugang zu den auf Azure verfügbaren umfangreichen Rechenressourcen zu schaffen. Ein Hauptvorteil der Durchführung von ML-Datenabfragen in Azure besteht darin, dass Sie auf eine große Menge an Rechenleistung zugreifen und diese nur für die Zeit nutzen können, die zur Durchführung Ihrer Berechnungen benötigt wird. In traditionellen Umgebungen würden diese teuren Rechenressourcen für große Zeiträume zwischen den Datenverarbeitungsaufträgen ungenutzt bleiben.

Eine weitere coole Ressource, die Sie bei der Durchführung von ML und Number Crunching auf Azure unterstützt, sind Data Science Virtual Machines (DSVMs). Diese VMs sind sowohl für Linux als auch für Windows verfügbar. Sie werden mit vielen gängigen Anwendungen vorinstalliert geliefert, darunter Jupyter Notebooks, Anaconda Python und R Server oder SQL Server. Weitere finden Sie in Abbildung 17.5.

Abbildung 17.5 DSVMs sind sowohl für Linux als auch für Windows verfügbar. Dieser Windows Server 2016 DSVM wird mit mehreren vorinstallierten Datenwissenschaftsanwendungen wie R-Server und Jupyter Notebooks geliefert. Mit DSVMs können Sie schnell mit der Verarbeitung großer Datenmengen und dem Aufbau von ML-Algorithmen beginnen.

Es ist nicht erforderlich, alle Tools und Abhängigkeiten auf Ihrem lokalen Computer zu installieren – Sie können eine DSVM mit so vielen CPU- und Speicherressourcen erstellen, wie Sie benötigen, um Ihre Daten schnell zu verarbeiten, und dann die VM löschen, wenn Ihr Verarbeitungsauftrag abgeschlossen ist und Sie die Datenmodelle haben, die Sie benötigen.

17.2 Azure Cognitive Services

Okay, was ist dann mit KI-Diensten, die Ihre Apps intelligenter machen? Bei Azure bilden eine Reihe von verwandten Diensten die Cognitive Services Suite. Die Dienste decken einige gemeinsame Bereiche der KI ab, mit denen Sie diese intelligenten Ressourcen schnell in Ihre Anwendungen integrieren können. Diese sind in die folgenden allgemeinen Bereiche unterteilt:

- Vision
- Spracherkennung
- Sprache
- Entscheidung
- Suche

Mehr als zwei Dutzend Dienste sind Teil der Cognitive Services-Familie. Manche dieser Dienste sind

- *Vision*, was Folgendes umfasst
  - *Computer Vision* für die Bildanalyse, Untertitelung und Tagging.
  - *Face* für die Analyse und Erkennung von Gesichtern in Bildern.
- *Speech*, was Folgendes umfasst
  - *Speech Services*, um Sprache zu analysieren und in Text umzuwandeln
  - *Lautsprechererkennung* zum Identifizieren und Überprüfen des Lautsprechers.
- *Language*, was Folgendes umfasst
  - *Language Understanding (LUIS)* , um die Interaktion mit Benutzer*innen zu verstehen und zu verarbeiten. Am Ende dieses Kapitels erkunden wir LUIS in einer Übung.
  - *Translator Text* für die Analyse und Korrektur von Rechtschreibfehlern oder die Durchführung von Übersetzungen.
- *Decision*, was Folgendes umfasst
  - *Content Moderator*, um Fotos, Videos und Texte zu überprüfen und zu bearbeiten
  - *Personalizer* für die Analyse von Mustern und die Bereitstellung von Empfehlungen für Kund*innen
- *Search*, was Folgendes umfasst
  - *Bing Custom Search* , um die Suche für Ihre benutzerdefinierten Daten und innerhalb von Anwendungen zu implementieren
  - *Bing Autosuggest*, um automatische Vorschläge zu machen, wenn Benutzer*innen Suchbegriffe und Anfragen eingeben

Wie Sie sehen können, kombinieren viele Azure-Dienste AI- und ML-Funktionen. Dieses Kapitel konzentriert sich auf die Sprache, insbesondere LUIS. Dieser Service wird häufig verwendet, um einen intelligenten Bot zu erstellen, der Kund*innen auf Ihrer Website helfen kann. Sie können dann eine Anwendung erstellen, die KI-Dienste in Azure verwendet und Sätze und Fragen interpretiert und die passende Antwort liefern kann, um einen Benutzer durch einen Bestellvorgang oder eine Supportanfrage zu führen.

17.3 Erstellung eines intelligenten Bots zur Unterstützung bei Pizzabestellungen

Ein *Bot* ist eine Anwendung, die programmiert ist, um auf Aufgaben und Eingaben eines Benutzenden zu reagieren. Wenn das so ziemlich nach einer normalen Anwendung klingt, dann liegt das daran, dass es eine ist! Der Unterschied besteht darin, wie die Botanwendung die Antwort bestimmt.

Ein grundlegender, allgemeiner Bot ist oft nichts anderes als eine Anwendung, die eine gewisse Form der Automatisierung bietet. Wenn ein*e Benutzer*in eine Nachricht sendet, ein Tag auf eine E-Mail-Nachricht setzt oder einen Suchbegriff sendet, führt der Bot vorprogrammierte Aufgaben aus, die eine bestimmte Aktion ausführen. Es gibt hier keine echte KI oder ML. Die Bot-Anwendung reagiert nur auf Benutzungseingaben.

Mit dem richtigen Framework kann ein Bot erweitert werden und erhält etwas mehr Freiheit und Intelligenz. Zu Beginn unseres Überblicks über KI habe ich davon gesprochen, wie eine typische Anwendung mit allen erwarteten Benutzungseingaben vorprogrammiert werden muss und wie die entsprechende Ausgabe aussehen würde. Es gibt jedoch keine Flexibilität, wenn der Benutzende beispielsweise einen anderen Eingabesatz oder einen Rechtschreibfehler eingibt.

Microsoft entwickelt das Bot Framework, das es einem Azure-Bot ermöglicht, die Bot Builder SDKs einfach zu integrieren und sich mit den Azure Cognitive Services zu verbinden. Mit minimaler Code-Erfahrung können Sie intelligente Bots erstellen, die die Leistung von Azure nutzen, um eine großartige Customer Experience zu bieten. Versuchen Sie nur nicht, Skynet zu bauen, es sei denn, Sie erinnern sich, wie *Terminator* ausging!

17.3.1 Erstellen eines Azure Web App Bots

Setzen wir einen Bot ein und integrieren einige KI- und ML-Dienste. Der Bot läuft in einer Azure-Webanwendung und verwendet das Microsoft Bot Framework, um sich mit LUIS zu verbinden und einem Kunden die Bestellung von Pizza zu ermöglichen. Abbildung 17.6 zeigt, was diese Übungen aufbauen und welche Dienste genutzt werden.

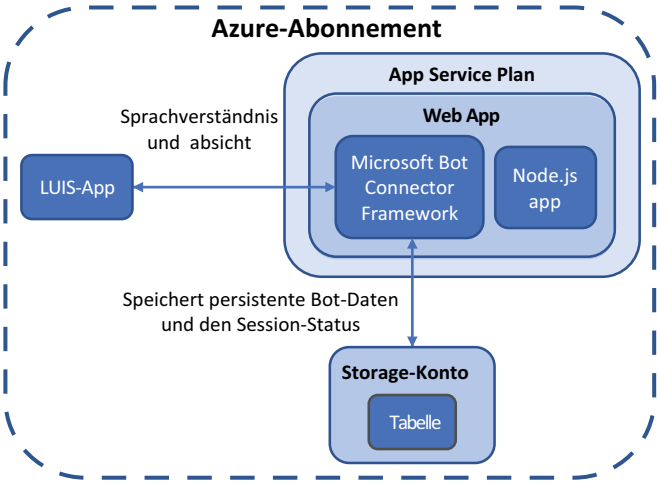


Abbildung 17.6 In den kommenden Übungen erstellen Sie einen Web App Bot, der mehrere Azure KI- und ML-Dienste integriert, um mit einem oder Kund*in zu interagieren und ihm zu helfen, Pizza zu bestellen.

Jetzt testen

Um einen Azure Web App Bot zu erstellen, führen Sie die folgenden Schritte aus:

1. Öffnen Sie das Azure Portal, und wählen Sie oben links in der Ecke „Create a Resource“ (Ressource erstellen) aus.
2. Suchen Sie Web App Bot aus und klicken Sie dann auf „Create“ (Erstellen).
3. Geben Sie einen Namen für Ihren Bot an, z. B. `azuremol`, ein, und erstellen Sie dann eine neue Ressourcengruppe zum Beispiel mit dem Namen `azuremolchapter17`.
4. Wählen Sie die für Sie am besten geeignete Region aus, und wählen Sie dann die F0-Preisstufe. Ihr Bot wird nicht viele Nachrichten verarbeiten, also ist die kostenfreie (F0) Stufe in Ordnung.
5. Wählen Sie eine Bot-Vorlage und die Sprache Node.js SDK.
6. Erstellen Sie einen einfachen Bot, da wir in einer späteren Übung unseren eigenen Beispielanwendungscode bereitstellen. In diesem Schritt wird eine LUIS-App erstellt, mit der Sie Sprachtraining und ML durchführen können.
7. Wählen Sie die am besten passende Region für Ihre LUIS-App, und erstellen Sie ein neues LUIS-Konto.
8. Geben Sie einen Namen für das LUIS-Konto an, z. B. `azuremol`. Dieser LUIS-Account verwaltet das User Sentiment für unseren Bot.
9. Wählen Sie „App Service Plan“ (App Service-Plan), und erstellen Sie einen neuen Plan. Geben Sie einen Namen an, z. B. `azuremol`, und wählen Sie erneut die für Sie am besten geeignete Region aus.
10. Deaktivieren Sie App Insights, da Ihr Bot sie nicht verwenden wird. Wie in früheren Kapiteln über Web Apps können Sie für die Leistungsfähigkeit von App Insights für die Produktion nutzen, um einen Einblick in die Leistung Ihrer Anwendung zu erhalten, indem Sie Daten und Analysen direkt aus dem Code streamen.
11. Akzeptieren Sie die Option zur automatischen Erstellung der Microsoft-App-ID und des Kennworts, akzeptieren Sie dann die Vereinbarung, und wählen Sie „Create“ (Erstellen).

Die Erstellung des Web App Bots und der zugehörigen Komponenten dauert wenige Minuten. Hinter den Kulissen passiert viel:

- Ein Azure App Service Plan wird erstellt.
- Es wird eine Webanwendung zusammen mit einer Node.js-Beispielswebanwendung bereitgestellt.
- Eine LUIS-App wird erstellt, und die Verbindungsschlüssel werden mit Ihrer Webanwendung konfiguriert.
- Ein Bot wird mit dem Microsoft Bot Connector erstellt, und die Verbindungsschlüssel werden über Ihre Webanwendung konfiguriert.

17.3.2 Sprache und Absichtsverständnis mit LUIS

Einer der Azure Cognitive Service-Bereiche, die wir zuvor betrachtet haben, ist die Sprache. Dies ist sinnvoll, da oft eine bestimmte Form der Sprache verwendet wird, um mit einer KI zu interagieren. Mit LUIS können Sie eine Nachricht oder einen Satz des oder der Benutzer*in bearbeiten und dessen Absicht bestimmen. Diese Absicht hilft dann Ihrer App, eine angemessene Antwort zu geben. Erweitern wir Ihren Bot mit LUIS.

Jetzt testen

Um eine LUIS-App zu erstellen und mit ML zu trainieren, führen Sie die folgenden Schritte aus.

1. Öffnen Sie in einem Webbrowser www.luis.ai, und melden Sie sich mit den gleichen Microsoft-Anmeldedaten wie für ihr Azure-Abonnement an.
2. Wählen Sie „My Apps“ (Meine Apps), und wählen Sie Ihre App, z. B. `azuremol`. Ihr LUIS-Anwendungsname enthält wahrscheinlich einige zusätzliche numerische Zeichen, die von dem Bot-Namen, den Sie im Azure Portal angegeben haben, abgeleitet sind.

   Es wurden einige vorgefertigte Absichten erstellt, aber Sie möchten die LUIS-App mit einem Beispiel überschreiben, das mehr auf die Pizzeria ausgerichtet ist.
3. Laden Sie zunächst die Datei azuremol.json von GitHub unter https://github.com/fouldsy/ azure-mol-samples-2nd-ed/blob/master/17/luisapp/azuremol.json auf Ihren lokalen Computer herunter. Um es einfacher zu machen, wählen Sie die Schaltfläche „Raw“ in GitHub, um nur den Inhalt der Datei anzuzeigen.
4. Zurück in ihrer LUIS-APP, wählen Sie die Verwaltung der App und dann „Versions“ (Versionen) aus.
5. Wählen Sie die Option zum Importieren einer Version aus, suchen Sie nach der heruntergeladenen Datei azuremol.json, und wählen Sie diese aus. Geben Sie einen Versionsnamen wie `1.0` ein, und wählen Sie dann „Done“ (Fertig).
6. Gehen Sie zurück zu „Build“ (Entwicklung) im oberen Menü, um die importierten Absichten der Beispielanwendung zu sehen. Wählen Sie einen oder zwei der Absichten, wie z. B. `greetings oder orderFood`, und schauen Sie sich einige der Beispielsätze an, die ein Kunde verwenden könnte, um mit dem Bot zu kommunizieren.
7. Bevor Sie die App in Aktion sehen können, müssen Sie sie trainieren. Wählen Sie „Train“ (Trainieren), und warten Sie dann einige Sekunden, bis der Prozess abgeschlossen ist. Abbildung 17.7 zeigt die ML-Prozesse beim Training Ihrer LUIS-Anwendung an.

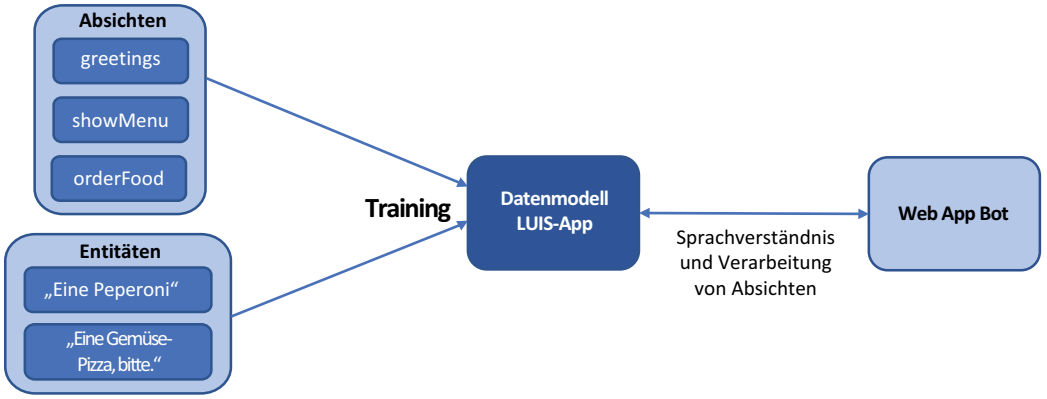


Abbildung 17.7 Wenn Sie die LUIS-App trainieren, werden die Absichten und Entitäten eingegeben und verarbeitet, um ein Datenmodell zu erstellen. Dieses Datenmodell wird dann von Ihrem Web App Bot verwendet, um das Sprachverständnis und die Absicht zu verarbeiten. Die Anzahl von zur Verarbeitung eingegebenen Absichten und Entitäten ist gering, sodass das Datenmodell nicht perfekt ist. In der realen Welt würden viel mehr Absichten und Entitäten bereitgestellt, und Sie würden das Datenmodell wiederholt trainieren, testen und verfeinern, um nach und nach größere Datensätze zu erstellen und ein genaues Modell für die Verarbeitung von Sprache und Absicht zu erstellen.

In einer komplexeren, realen Anwendung kann es länger dauern, bis dieser Trainingsprozess abgeschlossen ist, da alle Ihre Absichten und Entitäten von den

ML-Algorithmen verarbeitet werden, um das erforderliche Datenmodell für Ihre Anwendung zu erstellen, das angemessen auf die Kundenkommunikation reagieren kann.

8 Wenn die LUIS-App trainiert ist, wählen Sie „Test“ (Testen), und geben Sie ein paar Begrüßungen ein, z. B. *Hi* und *Hallo*. Unter jeder Ihrer Nachrichten befindet sich die am höchsten bewertete Absicht, zusammen mit der Wahrscheinlichkeit, dass die Nachricht oder Äußerung, die Sie eingegeben haben, mit der Absicht übereinstimmt. Diese grundlegenden Begrüßungen sollten zur Absicht `Begrüßung` passen.

9 Versuchen Sie, eine andere Begrüßung einzugeben, wie z. B. *(Guten) Tag* oder *(Guten) Abend*. Die Begrüßung aus einem Wort, basierend auf der Tageszeit kann zu einer falschen, obersten Absicht führen, wie z. B. `orderStatus`. Probieren Sie einige andere Sätze aus, bis etwas nicht mehr mit der erwarteten Absicht übereinstimmt. Das deutet dann darauf hin, dass die LUIS-App nicht ganz versteht, was Sie meinen. Wählen Sie eine Ihrer falschen Nachrichten aus, wie z. B. *Morgen*, und wählen Sie „Inspect“ (Überprüfen).

10 Wählen Sie im Menü „Inspect“ (Überprüfen) die Option zum Bearbeiten der falschen, obersten Absicht. Wählen Sie im Dropdown-Menü die Option `Begrüßung` oder was auch immer die am besten geeignete Absicht für Ihren falschen Satz ist.

11 Sie haben eine Änderung an Ihrer App vorgenommen, also wählen Sie erneut das Training der LUIS-App. Abbildung 17.8 zeigt, wie Sie zusätzliche Eingaben für die ML-Algorithmen bereitstellen, um dann das Datenmodell zu verarbeiten und das Sprachverständnis und die Absicht zu verfeinern.

12 Geben Sie im Fenster für Testmeldungen die falsche Meldung erneut ein, z. B. *Morgen*. Diesmal sollte die oberste Absicht korrekt identifiziert werden als `Begrüßung`.

13 Um die aktualisierte LUIS-App Ihrem Web App Bot zur Verfügung zu stellen, wählen Sie im oberen Menü die Option zum Veröffentlichen. Akzeptieren Sie alle Standardeinstellungen und wählen Sie die Veröffentlichung in einem Produktionsslot. Es dauert einige Sekunden, bis der Veröffentlichungsprozess abgeschlossen ist.

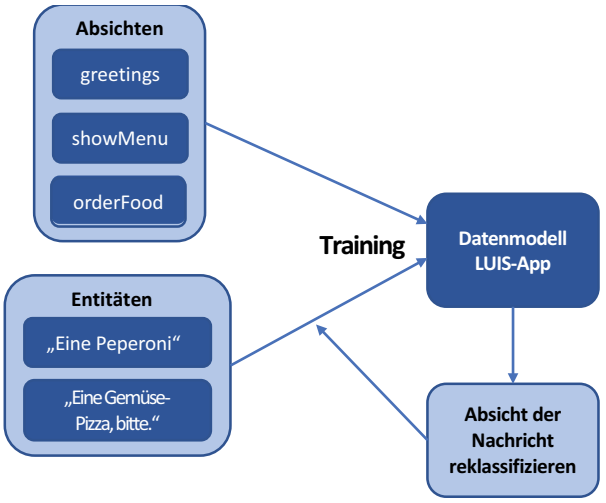


Abbildung 17.8 Während Sie die Absicht von Nachrichten neu klassifizieren und die LUIS-App neu trainieren, wird das Datenmodell verfeinert, da den ML-Algorithmen zusätzliche Dateneingaben zur Verfügung gestellt werden. Wenn Sie in Zukunft ähnliche Begrüßungen eingeben, wird das Datenmodell idealerweise verbessert und reagiert passender.

Denken Sie daran, dass Ihr Bot auf einer Web App läuft. Deswegen hat er Produktions- und Stagingslots, wie Sie in Kapitel 3 erfahren haben. In der realen Welt sollten Sie für einen Stagingslot veröffentlichen, überprüfen, ob alles wie erwartet funktioniert, und dann für den Produktionsslot veröffentlichen. Dieselben PaaS-Funktionen, mit denen Sie Webcode zwischen Entwicklungs- und Produktionslebenszyklen testen und verschieben konnten, kommen auch dem Lebenszyklus Ihres Web App Bots mit LUIS zugute.

Das grundlegende Beispiel zeigt, dass ML in der Lage war, Ihre Dateneingabe von *(Guten) Morgen* als Begrüßung aufzufassen und zu verstehen, dass ähnliche Grüße, wie z. B. *(Guten) Abend* auch Begrüßungen sind. ML funktioniert am besten, wenn ein großer Datensatz in das Datenmodell eingegeben werden kann. Daher ist es wichtig, dass Sie Ihre App gründlich testen und trainieren. Die KI, in diesem Fall die LUIS-App, ist nur so gut wie die Größe und Qualität der dem ML-Algorithmus zur Verfügung gestellten Daten.

17.3.3 Erstellen und Betreiben eines Web App Bots mit LUIS

Sie haben jetzt einen einfachen Web App Bot auf Azure und eine LUIS-App, die die Sprachverarbeitung übernimmt und die Kundenabsicht zurückgibt. Um die beiden zu integrieren, muss der Code für Ihren Bot geändert werden, damit er LUIS verwenden kann. SDKs sind für die Programmiersprachen C# und Node.js verfügbar. Ich finde, dass Node.js es ein wenig schneller und einfacher macht zu verstehen, was im Code passiert, wenn das alles neu für Sie ist. Wenn Sie mit C# vertraut sind, können Sie gerne C#-SDK erkunden, wenn Sie mit diesem Kapitel fertig sind. Fürs Erste verwenden wir eine einfache Node.js-App aus dem GitHub-Repository, um Ihren Bot in Aktion mit LUIS zu sehen.

Jetzt testen

Um Ihren Web App Bot mit Ihrem geschulten LUIS-Bot zu aktualisieren, führen Sie die folgenden Schritte aus:

1. Wählen Sie im Azure Portal links im Menü Ressourcengruppen aus und wählen Sie dann Ihre Ressourcengruppe, z. B. `azuremolchapter17`, aus. Wählen Sie dann Ihren Web App-Bot, z. B. `azuremol`.

   Verwenden wir einen Beispiel-Bot aus unserem GitHub-Repository. Der Beispiel-Bot ist in Node.js geschrieben, aber wie bei früheren Beispiel-Apps gilt: Keine Sorge, wenn das nicht Ihr Ding ist.

2. Um den Beispiel-Bot bereitzustellen, öffnen Sie die Cloud Shell. Klonen Sie gegebenenfalls die GitHub-Repository-Beispiele wie folgt in Ihrer Cloud Shell:

   ```
   git clone https://github.com/fouldsy/azure-mol-samples-2nd-ed.git
   ```

3. Wechseln Sie in das Verzeichnis für Kapitel 17:

   ```
   cd azure-mol-samples-2nd-ed/17/webappbot
   ```

4. Initialisieren Sie das Git Repository, und fügen Sie die Bot-Dateien hinzu:

   ```
   git init && git add . && git commit -m "Pizza"
   ```

5. Um den Beispiel-Bot hochzuladen, erstellen Sie eine Verbindung zu Ihrer Web App. Der folgende Befehl ruft das Web App-Repository ab und konfiguriert das Git-Repository Ihrer lokalen Samples zur Herstellung einer Verbindung. In den vorherigen Kapiteln habe ich Sie Ihnen die Aufgabe gegeben, nach dieser

Adresse zu suchen. Inzwischen hoffe ich, dass Sie sich bereits aktiv mit weiteren von Azure CLI beschäftigt haben und festgestellt haben, dass viele dieser Informationen leicht zugänglich sind:

```
git remote add webappbot \
  $(az webapp deployment source config-local-git \
  --resource-group azuremolchapter17 \
  --name azuremol \
  --output tsv)
```

6 Verschieben Sie den Node.js-Beispiel-Bot mit dem folgenden Befehl zu Ihrer Webanwendung:

```
git push webappbot master
```

7 Geben Sie bei Aufforderung das von Ihnen erstellte Kennwort für den Git-Benutzenden ein, den Sie in früheren Kapiteln verwendet haben (das in Kapitel 3 erstellte Konto).

Für den Fall, dass Sie Ihr Git-Kennwort nicht auf einer Haftnotiz notiert haben

Wenn Sie das Kennwort vergessen haben, können Sie es zurücksetzen. Als Erstes erhalten Sie den Benutzungsnamen Ihres lokalen Git-Bereitstellungskontos:

```
az webapp deployment user show --query publishingUserName
```

Um das Kennwort zurückzusetzen, geben Sie den Namen Ihres Kontos aus dem vorherigen Befehl ein. Folgen Sie dann die Aufforderungen, um ein neues Kennwort festzulegen. Das folgende Beispiel setzt das Kennwort für das Benutzungskonto namens azuremol zurück:

```
az webapp deployment user set --user-name azuremol
```

Betrachten wir Abbildung 17.9, um zu sehen, was Sie bereitgestellt haben. Die LUIS-Anwendung ist nun mit ML-Algorithmen trainiert, und Ihr Datenmodell ist bereit für die Node.js-App, damit Kund*innen interagieren und Pizza bestellen können.

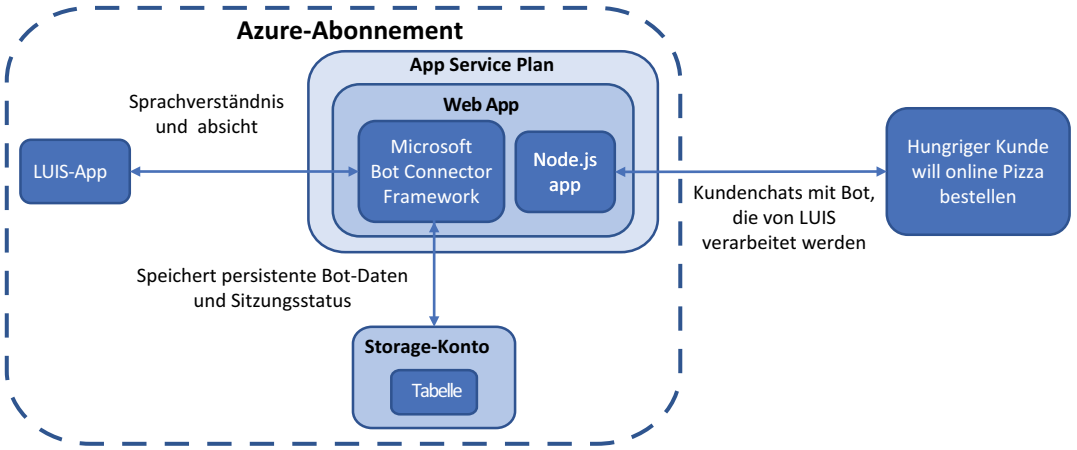


Abbildung 17.9 Ein Kunde kann nun online auf Ihren Bot zugreifen und darum bitten, sich die Speisekarte anzusehen oder Pizza zu bestellen. LUIS bietet das Sprachverständnis, das es dem Bot ermöglicht, Aufträge zu bearbeiten und zur weiteren Verarbeitung an Azure Storage zu senden.

Zurück im Azure Portal für Ihren Web App Bot, wählen Sie die Option „Test“ (Testen) in Web Chat. Es dauert einige Sekunden, wenn Sie sich zum ersten Mal mit dem Bot verbinden, aber Sie sollten dann in der Lage sein, zu interagieren, die Liste der Pizzas in der Speisekarte anzuzeigen und eine Reihenfolge zu erstellen, wie in Abbildung 17.10 dargestellt. Probieren Sie es selbst aus!

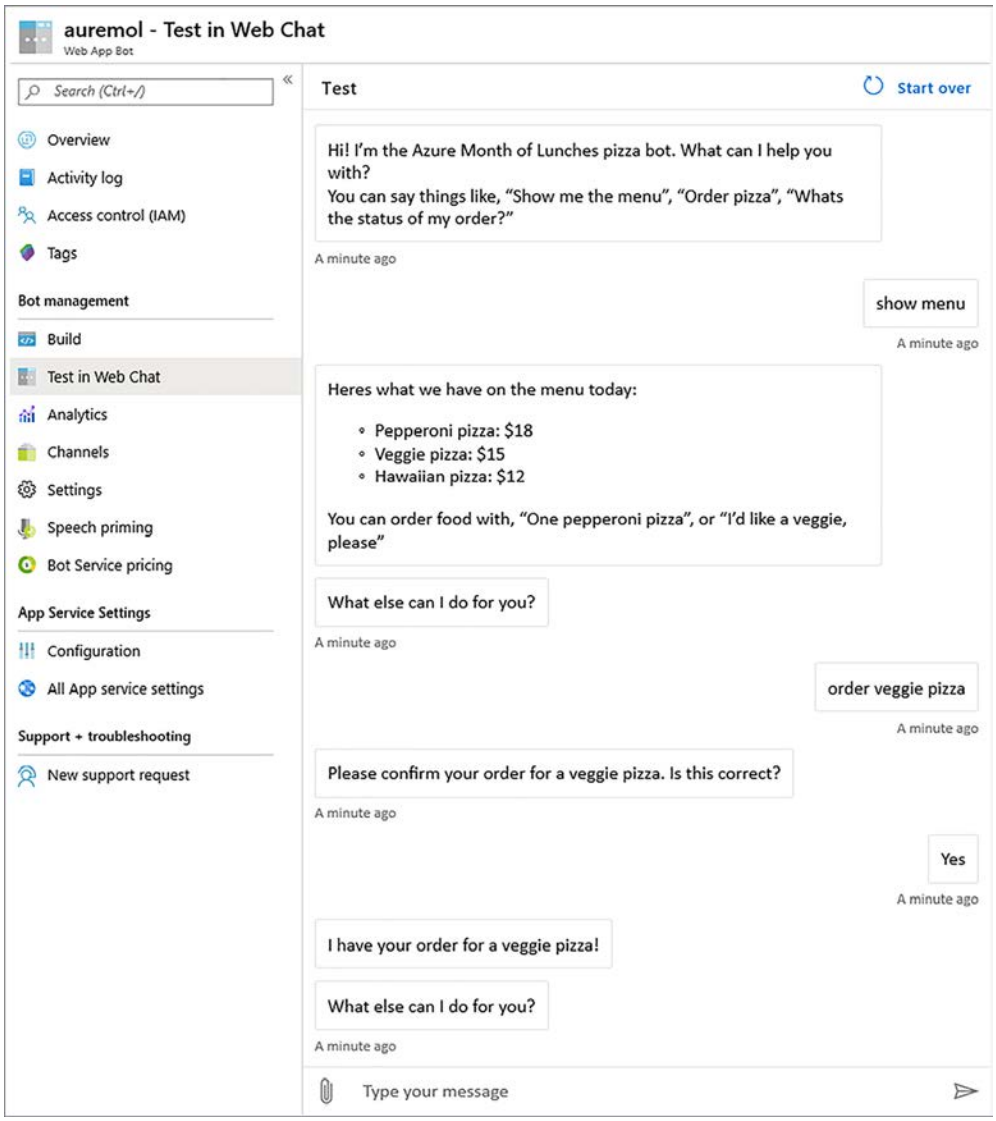


Abbildung 17.10 Wenn Ihr Web App Bot läuft, beginnen Sie ein Gespräch und versuchen Sie, eine Pizza zu bestellen. In diesem Beispielsdialog können Sie sich die Speisekarte ansehen, eine Pizza bestellen und den Bestellstatus überprüfen. Die App ist einfach und erstellt nicht wirklich Bestellungen oder aktualisiert den Status über die Art der Pizza hinaus, die bestellt wurde, aber die Übung zeigt Ihnen hoffentlich, wie man einen Bot schnell in Azure einsetzen kann.

Ich hoffe, dass diese grundlegenden Übungen Ihnen eine Vorstellung davon vermittelt haben, was Azure für KI und ML bieten kann. Der Web App Bot mit LUIS kann um zusätzliche Azure Cognitive Services wie Spell Check und Translator erweitert werden. Mit diesen Diensten können Sie Wörter und Sätze interpretieren, wenn der oder die Benutzer*in sie falsch schreibt, oder Ihren Bot in mehreren Sprachen sprechen lassen. Oder Sie könnten Face and Personalizer verwenden, um anhand der Gesichtserkennung der Kamera zu erkennen, welcher Kunde eine Bestellung aufgegeben hat, und ihm automatisch Pizzen vorschlagen, die ihm gefallen könnten.

ML war Teil der LUIS-App, aber es gibt noch viel mehr ML-Ressourcen und -Tools auf Azure. Die Möglichkeit, große Datensätze zu verarbeiten und ML-Datenmodelle auf leistungsstarken Azure-Compute-Ressourcen zu berechnen, senkt den Einstiegspunkt für Sie, Anwendungen zu erstellen, die durch einige wichtige Datensätze unterstützt werden. Die Anwendungen sind genauer und effizienter, und Sie müssen keine Hardware kaufen oder spezielle Tools zu installieren, da die DSVMs alle erforderlichen Komponenten enthalten. Nicht alle Anwendungen passen gut zu KI und ML, aber wenn Kund*innen mehr von dem erwarten, was Ihr Unternehmen bieten kann, können diese Azure-Dienste oft helfen, sich von der Konkurrenz abzuheben.

Batch-Verarbeitung von Workload

Zwei weitere Bereiche von Azure, die im Hinblick auf Big Data und Compute für ML von Interesse sein könnte, sind die Azure Batch- und HPC-Services. Mit Azure Batch können Sie große, sich wiederholende Compute-Aufgaben durchführen, ohne dass Sie haufenweise Planer für die Arbeit verwalten müssen. Batch führt Aufgaben auf VMs mit einem eigenen Management und Planer aus, um Ihnen zu helfen, so wie Skalierungsgruppen, die Autoskalierung und Lastausgleich für VMs beinhalten. Obwohl Batch nicht direkt mit ML zu tun hat, ist es eine gute Wahl, wenn Sie andere große Computing-Processing-Aufgaben haben.

Es gibt auch High-Performance-Computing (HPC)-Komponenten in Azure für große VMs oder den Zugriff auf Grafikprozessor (GPU) VMs. Spezifische Tools und Suiten wie DataSynapse und Microsoft HPC Pack können auch zum Ausführen von Anwendungen verwendet werden, die eine große Rechenleistung erfordern.

Bereiche wie ML, Azure Batch und HPC sind gute Beispiele dafür, wie man Cloud Computing-Anbieter wie Azure nutzen kann, um große Rechenaufgaben auszuführen. Sie bezahlen nur für die von Ihnen verwendeten Rechenressourcen, sodass Sie keine teuren Geräte, die Sie nur wenig nutzen, kaufen und warten müssen.

17.4 Übung: Hinzufügen von Kanälen für die Bot-Kommunikation

In den früheren Beispielen haben Sie mit Ihrem Bot über ein Testfenster im Azure Portal kommuniziert. Kanäle ermöglichen es Ihnen, die Interaktionen mit Ihrem Bot zu erweitern. Sie können Ihrem Bot erlauben, mit Skype oder Facebook Messenger oder Anwendungen wie Microsoft Teams und Slack zu kommunizieren. Der Azure Bot Service vereinfacht die Schritte, die erforderlich sind, um einen Bot in diese externen Dienste zu integrieren:

1. Wählen Sie im Azure Portal Ihren Web App Bot aus, und wählen Sie dann „Channels“ (Kanäle).
2. Wählen Sie einen Kanal, der Ihnen zusagt, wie z. B. Skype.

Andere Kanäle erfordern oft, dass Sie eine Entwicklungsverbindung herstellen, z. B. zu Facebook oder Slack. Mit Skype können Sie einen HTML-Code kopieren und einfügen, damit er funktioniert.

3. Geben Sie alle erforderlichen Informationen an, wie z. B. die Bot-Anwendungs-ID. Sie finden diese ID unter Einstellungen für die Bot-Verwaltung.
4. Verwenden Sie bei Bedarf den Online-Code-Editor, um eine einfache HTML-Seite, wie z. B. default.htm, im Verzeichnis wwwroot zu erstellen, und fügen Sie jeden eingebetteten Code für Ihren Kanal ein. Sie können Ihre Webanwendung über das Azure Portal öffnen und dann ihre URL auswählen, um die Seite default.htm zu öffnen, die Ihren Kanalcode enthält, wie z. B. http://azuremol.azurewebsites.net/default.htm.

18 Azure Automation

Nach Möglichkeit sollten Sie sich nicht manuell auf einem Server anmelden und Änderungen vornehmen. Software muss nicht durch Anklicken von Schaltflächen in einer GUI installiert werden, und es müssen keine Updates für Konfigurationsdateien in einem Texteditor vorgenommen werden. Diese manuellen Aktionen können zu Fehlern führen, die Fehlkonfigurationen und Anwendungsausfälle verursachen können. Können Sie sich an alle erforderlichen Schritte erinnern, wenn Sie die Konfiguration eines Servers replizieren möchten, um den bestehenden Server in Betrieb zu nehmen? Was ist, wenn Sie es in sechs Monaten erneut durchführen müssen?

In Kapitel 16 haben wir uns mit einer Möglichkeit beschäftigt, automatisch nach Updates zu suchen und diese auf Server anzuwenden. Dieser Zauber geschah mit dem Einsatz von Azure Automation. In diesem Kapitel untersuchen wir, wie Sie Runbooks erstellen, ausführen und bearbeiten können. Wir erläutern auch, wie Sie mit der der PowerShell Desired State Configuration Anwendungen automatisch installieren und Server konfigurieren können.

18.1 Was ist Azure Automation?

Ein Azure Automation-Konto vereint viele Elemente, wie in Abbildung 18.1 dargestellt. Ein Kernelement ist die Erstellung und Ausführung von Skripten On-Demand oder nach einem definierten Zeitplan. Sie können Skripte in PowerShell oder Python erstellen und die Planung und Ausführung dieser Runbooks der Azure-Plattform überlassen. Sie können Anmeldeinformationen und Verbindungsobjekte gemeinsam nutzen und automatisch die gewünschten Konfigurationen von Servern nutzen und über sie berichten. Das Update Management, das wir in Kapitel 16 untersucht haben, gewährleistet die Sicherheit Ihrer Server und hält sie auf dem neuesten Stand mit den neuesten Host-Patches und Updates während des gesamten Lebenszyklus Ihrer Anwendungsumgebung.

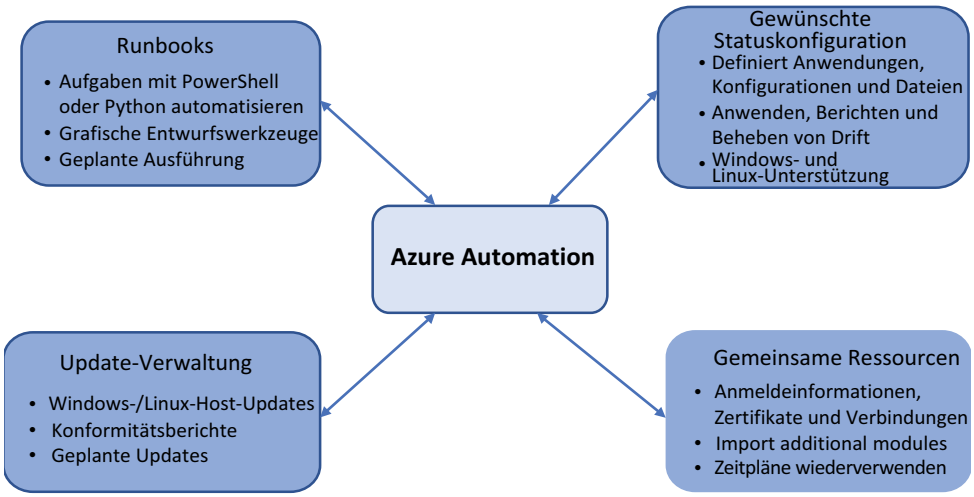


Abbildung 18.1 Azure Automation bietet viele verwandte Funktionen. Ein gemeinsamer Satz von Ressourcen, darunter Anmeldeinformationen, Zertifikate, Zeitpläne und Verbindungsobjekte, kann verwendet werden, um PowerShell- oder Python-Skripte automatisch auf Ziel-Servern auszuführen. Sie können den gewünschten Zustand eines Servers definieren, und Azure Automation installiert und konfiguriert den Server in gewünschter Weise. Host-Updates und Sicherheitspatches können automatisch installiert werden. Alle diese Funktionen funktionieren sowohl auf Windows- als auch auf Linux-Servern, auf Azure und On-Premises oder bei anderen Cloud-Anbietern.

Um die Verwaltung über mehrere Runbooks oder die gewünschte Statuskonfigurationen in einem Automation-Konto zu vereinfachen, können Sie die folgenden Ressourcen nutzen:

- *Mit Zeitplänen* können Sie eine Reihe von Zeitpunkten und Wiederholungen definieren, die auf jedes Runbook oder jeden Update Management-Auftrag angewendet werden können. Wenn Sie ein reguläres Ereignis später ändern möchten, können Sie einen der gemeinsamen Zeitpläne ändern, anstatt jeden einzelnen Runbook- oder Update Management-Auftrag, der sie verwendet, ändern zu müssen.
- *Module* erweitern die Kernfunktionalität durch die Speicherung zusätzlicher PowerShell-Module. Die Windows PowerShell- und Azure-Basismodule sind bereits verfügbar, aber zusätzliche Module, wie beispielsweise für die Linux-Verwaltung, können ergänzt und in Runbooks verwendet werden.
- *Anmeldeinformationen* für die verschiedenen Konten, die über die Berechtigung zum Ausführen verschiedener Runbooks verfügen, werden als Assets gespeichert und nicht in jedem Runbook definiert. Mit diesem Ansatz können Sie die Anmeldeinformationen bei Bedarf aktualisieren und zurücksetzen. Jedes Runbook, das diese Anmeldeinformationen verwendet, wird automatisch aktualisiert. Daher werden Anmeldeinformationen nicht als einfacher Text in Runbooks gespeichert, was die Sicherheit von Runbooks erhöht.
- *Verbindungen* definieren die Authentifizierungseigenschaften für Azure AD Service Principals. Dies ist ein spezieller Typ von Benutzungskonto, das Runbooks den Zugriff auf Ihre Azure-Ressourcen ermöglicht. Diese Verbindungen verwenden in der Regel digitale Zertifikate, und keine Benutzungsnamen und Kennwörter, um eine zusätzliche Sicherheitsebene zu schaffen.

- *Zertifikate* werden oft in Verbindungsassets integriert, um eine sichere Möglichkeit zur Überprüfung der Identität eines Service Principals zu schaffen. Wie bei den Basis-Anmeldeinformationen können Sie diese Zertifikate regelmäßig an einem zentralen Ort aktualisieren. Jedes Runbook, das diese Zertifikate verwendet, kann automatisch auf die neuen Zertifikate zugreifen. Sie können eigene Zertifikate zur Verwendung mit Runbooks oder Konfigurationsdefinitionen für gewünschte Zustände erstellen und speichern.
- *Variablen* stellen eine zentrale Stelle zur Speicherung von Laufzeitwerten wie Namen, Standortzeichenketten und ganzen Zahlen zur Verfügung. Diese Variablen werden bei der Ausführung Ihrer Runbooks injiziert. Dieser Ansatz begrenzt die Anzahl der hartkodierten Ressourcen in jedem Runbook.

Intelligenter statt härter arbeiten

In Kapitel 16 haben wir erläutert, wie Azure-Verwaltungsdienste zur Überwachung und Berichterstellung auf Servern auf Azure, On-Premises oder bei anderen Cloud-Anbietern zusammenarbeiten. Sie installieren und konfigurieren die erforderlichen Agents auf Remote-Servern und bieten ihnen dann eine Möglichkeit, sich wieder mit der Azure-Infrastruktur zu verbinden.

Azure Automation kann auch plattform- und infrastrukturübergreifend arbeiten. So kann beispielsweise der hybride Runbook-Worker Automation-Runbooks auf Servern außerhalb von Azure ausführen. Sie verwenden weiterhin die gemeinsamen Automation-Assets, die Anmeldeinformationen, Verbindungen und Zertifikate definieren, nur diesmal können diese Assets zur Definition der Authentifizierungskomponenten für verschiedene Plattformen eingesetzt werden. Sie können auch die gewünschten Zustandskonfigurationen auf Nicht-Azure-VMs sowohl für Windows als auch für Linux verwenden.

In jedem Fall wird eine Gateway-Komponente in der Remoteumgebung installiert, die als Proxy für die Automation-Befehle fungiert, während sie an die festgelegten Ziele gesendet werden. Dieser Gateway Proxy-Ansatz bietet einen zentralen Verbindungspunkt zur Automatisierung in Remote-Umgebungen und minimiert Sicherheitsbedenken, da es keinen direkten Zugriff auf anderweitige Remote Server bietet.

Runbooks und Definitionen der gewünschten Zustandskonfiguration müssen möglicherweise ein wenig überarbeitet werden, um auf physischen On-Premises-Servern im Vergleich zu Azure VMs zu laufen. Wie bei Azure Backup, Site Recovery oder Update Management besteht der Vorteil von Azure Automation darin, dass es eine zentrale Verwaltungsebene und eine Reihe von Tools bietet, um die Automatisierung über all Ihre verschiedenen Infrastrukturen und Server hinweg zu ermöglichen.

18.1.1 Erstellen eines Azure Automation-Kontos

Lassen Sie uns starten und ein Azure Automation-Konto erstellen und uns die standardmäßig enthaltenen Runbooks ansehen. Die Demo-Runbooks bieten einen großartigen Framework zum Erstellen eigener Runbooks. Sie können auch einen grafischen Editor nutzen, mit dem Sie Bausteine per Drag & Drop verschieben können, um Automation-Skripte zu generieren.

Jetzt testen

Führen Sie die folgenden Schritte aus, um ein Azure Automation-Konto und Beispiel-Runbooks zu erstellen:

1. Klicken Sie oben links im Azure Portal auf „Create a Resource“ (Ressource erstellen).
2. Suchen und wählen Sie „Automation“, und wählen Sie dann „Create“ (Erstellen).

Die Option Automatisierung und Steuerung erstellt auch einen Operations Management Suite (OMS)-Workspace, wie in Kapitel 16 beschrieben, und konfiguriert den Automation Hybrid Worker für die Verwaltung von Ressourcen außerhalb von Azure. OMS ist ein wenig auf dem Rückzug und wird durch die zentralen Azure-Dienste ersetzt, die wir in den vorherigen Kapiteln betrachtet haben. Wählen Sie zunächst, ob Sie nur die Automatisierungsressource erstellen möchten.

3. Geben Sie einen Namen, z. B. azuremol, ein und erstellen Sie dann eine neue Ressourcengruppe, zum Beispiel `azuremolchapter18`.
4. Wählen Sie die am besten geeignete Azure-Region in Ihrer Nähe aus, und akzeptieren Sie die Option auf „Create Azure Run As Account“ (Azure als Konto ausführen erstellen).

   Mit der Option „Create Run As Account“ (Als Konto ausführen erstellen) erstellt zusätzliche Konten in Azure AD. Sicherheitszertifikate werden auch erstellt, damit sich die Konten automatisch authentifizieren können, ohne dass Benutzungsanweisungen oder das Speichern eines Kennworts erforderlich sind. Sie können zusätzliche reguläre Kontoanmeldeinformationen, die als Automation-Asset definiert sind, erstellen und angeben, um eine genauere Kontrolle darüber zu erhalten, welche Konten für die Ausführung bestimmter Runbooks verwendet werden.

In Kombination mit RBAC, die wir in Kapitel 6 behandelt haben, können Sie spezifische Run As-Konten („Ausführen-als-Konten“) für Runbooks erstellen. Sie stellen einen begrenzten Satz von Berechtigungen bereit, die zur Ausführung der von jedem Runbook oder jedem Satz von Runbooks erforderlichen Aufgaben notwendig sind. Aus sicherheitstechnischer Sicht ermöglicht Ihnen dieser Ansatz die Prüfung und Kontrolle darüber, wie und wann diese Konten verwendet werden. Sie sollten es vermeiden, ein einziges Run As-Konto mit adminähnlichen Berechtigungen zu erstellen, da dieser Ansatz wenig Schutz vor Missbrauch bietet.

18.1.2 Azure Automation-Assets und -Runbooks

Das Azure Automation Konto, das Sie im Abschnitt 18.1.1 erstellt haben, enthält einige Beispiel-Runbooks. Sowohl PowerShell als auch Python-Beispiele sind verfügbar. Verbindungsassets und Zertifikate werden auch dem Automation-Konto für die erstellten Run As-Konten hinzugefügt. Lassen Sie uns diese gemeinsamen Verbindungsassets näher kennenlernen.

Jetzt testen

Führen Sie die folgenden Schritte aus, um die konfigurierten Assets und Beispiel-Runbooks anzuzeigen:

1. Wählen Sie im Azure Portal links die Ressourcengruppen aus, und wählen Sie dann Ihre Gruppe, z. B. `azuremolchapter18`. Wählen Sie dann Ihr Azure Automation-Konto, z. B. `azuremol`.
2. Wählen Sie im Menü auf der linken Seite unter „Shared Resources“ (Gemeinsame Ressourcen) die Option „Connections“ (Verbindungen) aus.
3. Wählen Sie AzureRunAsConnection aus, wie in Abbildung 18.2 gezeigt.
4. Wählen Sie Zertifikate aus dem Hauptmenü des Automation-Kontos unter „Shared Resources“ (Gemeinsame Ressourcen) und anschließend „AzureRunAsCertificate“ aus. Wie in Abbildung 18.3 dargestellt, stimmt der digitale Thumbprint mit RunAsConnection aus dem vorherigen Schritt überein.

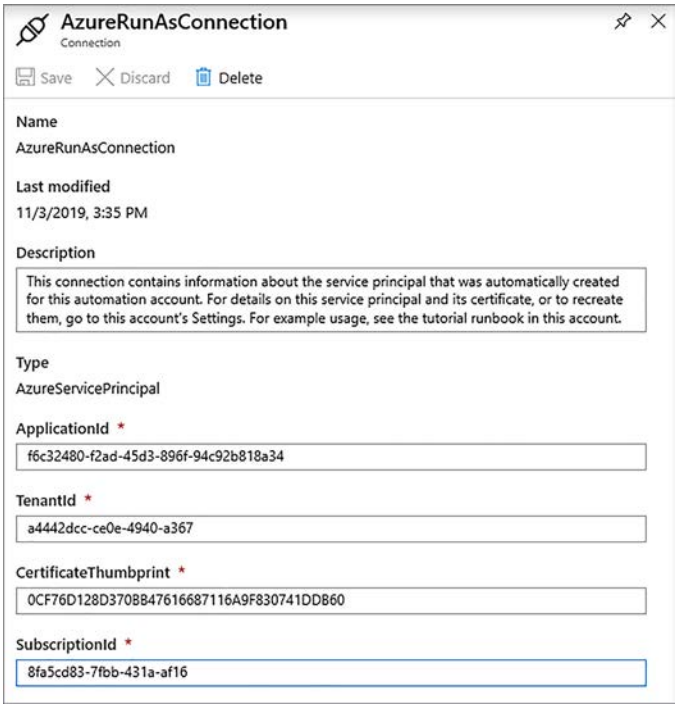


Abbildung 18.2 Zu den Informationen über das Konto „Run As“ (Ausführen als) gehören eine ApplicationId und TenantId-spezifische Eigenschaften für Azure AD, die zur Identifizierung der Anmeldeinformationen für dieses Konto beitragen. Ein CertificateThumbprint stimmt mit einem digitalen Zertifikat überein, mit dem wir uns im nächsten Schritt beschäftigen werden.

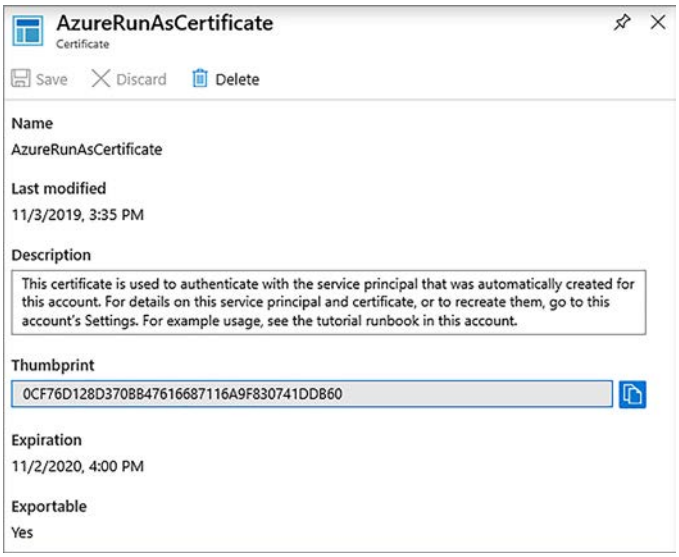


Abbildung 18.3 Der Thumbprint des RunAsCertificate stimmt mit dem in RunAsConnection angezeigten Thumbprint überein. In Ihren Runbooks legen Sie fest, welches Verbindungsasset verwendet werden soll. Zur Anmeldung am Azure Konto wird das entsprechende Zertifikat verwendet.

5 Nachdem Sie nun die Funktionsweise von Assets für Verbindungen und Zertifikate verstanden haben, lassen Sie uns einen Blick auf eines der Beispiel-Runbooks werfen. Wählen Sie Runbooks aus dem Menü links im Automation-Konto aus. Einige Beispiel-Runbooks sind verfügbar.

6 Wählen Sie das PowerShell-Runbook AzureAutomationTutorialScript.

7 Oben im Beispiel-Runbook finden Sie Optionen zum Starten, Anzeigen und Bearbeiten des Runbooks. Diese Optionen sollten selbsterklärend sein!

Sie haben auch eine Option für Zeitplanung, mit der Sie eine freigegebene Ressource erstellen oder auswählen können, die einen Zeitplan für die Ausführung des Runbooks zu einem bestimmten Zeitpunkt definiert, und eine Option für „Webhook", mit der Sie eine Webhook-URL erstellen können, um das Runbook aus einem anderen Skript oder einer anderen Aktion auszuführen. Wählen Sie „View" (Anzeigen) aus.

Azure Automation und Quellcodesteuerung mit GitHub

Runbooks können in ein Quellcodeverwaltungssystem wie GitHub integriert werden. Einer der großen Vorteile eines Quellcodeverwaltungssystem für Ihre Runbooks ist die Möglichkeit, die Änderungsverwaltung zu dokumentieren und im Falle eines Problems auf frühere Versionen der Runbooks zurückzugreifen.

Jedes Mal, wenn Sie ein Azure Automation-Runbook speichern, wird der Quellcodesteuerung eine neue Version zugewiesen. Dazu müssen Sie den Runbook-Editor nicht verlassen, da die Azure-Plattform und das konfigurierte Quellcodeverwaltungssystem so konfiguriert sind, dass sie in beiden Richtungen arbeiten. Wenn Sie mit dem neuen Runbook auf ein Problem stoßen, können Sie auf eine frühere Version aus der Quellcodesteuerung zurückgreifen, mit der Aufträge ohne Verzögerung weiterlaufen. Im Anschluss können Sie sich mit Fehlerbehebung für die aktualisierte Version beschäftigen.

Die Verwendung der Quellcodesteuerung bietet auch eine Aufzeichnung der Art und des Zeitpunktes der Änderungen. Wenn Sie Ihre Runbooks überprüfen oder verstehen müssen, wie sich diese im Laufe der Zeit entwickelt haben, bieten Quellcodesteuerungssysteme eine gute Möglichkeit, um die Unterschiede bei jeder Revision zu erkennen.

18.2 Azure Automation-Beispiel-Runbook

Lassen Sie uns untersuchen, wie das PowerShell-Beispiel-Runbook, AzureAutomationTutorialScript, eine Verbindung zu Azure herstellt und Informationen über Ihre Ressourcen sammelt. Sie können uns auch mit dem Python-Beispiel-Runbook folgen, wenn Sie möchten, denn das Layout ist ähnlich aufgebaut. PowerShell und Python sind die einzigen Sprachen, die derzeit in Azure Automation-Runbooks unterstützt werden. Die folgende Auflistung richtet die Verbindungs-Anmeldeinformationen im Runbook ein.

Abbildung 18.1 Einrichtung von Anmeldeinformationen der Verbindung

```
$connectionName = "AzureRunAsConnection"
try
{
  # Get the connection "AzureRunAsConnection "
  $servicePrincipalConnection=Get-AutomationConnection -Name
➥$connectionName
```

Erstellt ein Objekt für $connectionName

Stellt die Verbindungsanfrage

Erstellt ein Service Principal-Objekt

```
    "Logging in to Azure..."
    Add-AzureRmAccount `
        -ServicePrincipal `
        -TenantId $servicePrincipalConnection.TenantId `
        -ApplicationId $servicePrincipalConnection.ApplicationId `
        -CertificateThumbprint
➥$servicePrincipalConnection.CertificateThumbprint
}
```

Meldet sich bei Azure an

Der Code beginnt mit der Erstellung eines Objekts für `$connectionName`. In der Übung „Jetzt testen" haben Sie gesehen, dass ein Standard-Verbindungsasset für AzureRunAsConnection erstellt wurde. Wenn Sie Ihre eigenen Runbooks erstellen, können Sie zusätzliche Run As-Konten und Verbindungsassets erstellen, um die Runbooks und die von ihnen verwendeten Anmeldeinformationen zu trennen. Die Verbindungsteile und die Ausnahmebehandlung, die wir uns als Nächstes ansehen, sollten für alle Runbooks gleich sein. Bei Bedarf können Sie das zu verwendende Run As-Verbindungsasset ändern.

Als nächstes wird eine `Try`-Anweisung verwendet, um die Verbindungsanfrage zu stellen. Ein Service Principal mit dem Namen `$servicePrincipalConnection` wird basierend auf `$connectionName` erstellt. Das Runbook meldet sich dann in Azure mit `Add-AzureRmAccount` an und nutzt das Objekt `$servicePrincipalConnection`, um die `TenantId`, `ApplicationId` und `Certificate-Thumbprint` zu erhalten. Wir haben diese Parameter als Teil des Verbindungsassets bereits früher erläutert. Das Zertifikatasset, das mit dem Thumbprint von `$servicePrincipalConnection` übereinstimmt, wird verwendet, um die Anmeldung bei Azure abzuschließen.

Die nächste Auflistung zeigt, dass beim Fehler der Verbindung das Runbook den Fehler abfängt und die Ausführung unterbricht.

Auflistung 18.2 Behebung eines Fehlers und Unterbrechung der Runbook-Ausführung

```
catch {
     if (!$servicePrincipalConnection)
     {
          $ErrorMessage = "Connection $connectionName not found."
          throw $ErrorMessage
     } else{
          Write-Error -Message $_.Exception
          throw $_.Exception
     }
}
```

Die `Catch`-Anweisung bearbeitet alle Fehler im Rahmen des Anmeldeversuchs. Wenn eine Service Principal-Verbindung nicht gefunden werden konnte, wird ein Fehler ausgegeben. Dieser Fehler bedeutet in der Regel, dass das von Ihnen angegebene Verbindungsasset nicht gefunden werden kann. Überprüfen Sie noch einmal den Namen und die Schreibweise Ihrer Verbindung.

Andernfalls wurde das Verbindungsobjekt gefunden und der Service Principals zum Anmelden verwendet, aber dieser Authentifizierungsprozess war nicht erfolgreich. Dieser Fehler kann darauf zurückzuführen sein, dass das Zertifikat nicht mehr gültig ist oder das Run As-Konto nicht mehr aktiviert ist. Diese Funktion zeigt, wie Sie ein Konto in Azure AD widerrufen und sicherstellen können, dass alle Runbooks, die die Anmeldeinformationen verwenden, nicht mehr erfolgreich ausgeführt werden können.

Nun erhält das Runbook eine Liste aller Azure-Ressourcen.

Auflistung 18.3 Erhalt einer Liste von Azure-Ressourcen

```
$ResourceGroups = Get-AzureRmResourceGroup
foreach ($ResourceGroup in $ResourceGroups)
{
    Write-Output ("Showing resources in resource group "
➥+ $ResourceGroup.ResourceGroupName)
     $Resources = Find-AzureRmResource -ResourceGroupNameContains
➥$ResourceGroup.ResourceGroupName |
➥Select ResourceName, ResourceType
     ForEach ($Resource in $Resources)
     {
         Write-Output ($Resource.ResourceName + " of type "
➥+  $Resource.ResourceType)
     }
     Write-Output ("")
}
```

Der letzte Teil des Runbooks ist der Ort, an den Ihr Runbook-Code weitergeleitet werden würde. Ein Objekt wird für `$ResourceGroups` erstellt, das eine Liste aller verfügbaren Azure-Ressourcengruppen erhält. Eine `foreach`-Schleife durchläuft dann jede Ressourcengruppe, findet eine Liste von Ressourcen und schreibt eine Liste der Ressourcennamen und -Typen heraus.

Dieses einfache Beispiel zeigt, wie Sie mit Azure interagieren können, sobald sich das Runbook gegenüber dem Abonnement authentifiziert hat. Wenn Sie RBAC auf dem Run As-Konto implementieren, werden nur die Ressourcengruppen zurückgegeben, für die das Konto die Berechtigungen besitzt. Dieser Ansatz für RBAC zeigt, warum es ein guter Sicherheitsprinzipal ist, um Run As-Konten zu erstellen und zu verwenden, die den Zugriff der Runbooks auf Ressourcen in Ihrer Azure-Umgebung beschränken. Versuchen Sie immer, die geringstmögliche Anzahl von Berechtigungen bereitzustellen.

Machen Sie sich keine Sorgen, wenn all diese PowerShell- oder Python-Informationen Neuland für Sie sind. Beide bieten eine großartige, grundlegende Skriptsprache, können aber auch zur Entwicklung komplexer, leistungsstarker Anwendungen verwendet werden. Als Entwickler*in sollte jede der beiden Sprachen für Sie relativ einfach zu erlernen und zu bedienen sein. Wenn Sie ein IT-Profi sind, dann können Sie durch die Automatisierung von Aufgaben mehr Zeit für die Ausführung aller anderen, gestapelten Aufträge gewinnen. Sowohl PowerShell als auch Python bieten hier gute Ausgangspunkte. Manning Publications hat noch ein paar weitere großartige Bücher mit unterstützenden Informationen für Sie auf Lager!

18.2.1 Ausführen und Anzeigen von Ausgaben aus einem Beispiel-Runbook

Nachdem Sie nun die Inhalte des Beispiel-Runbook-Skripts kennenlernen konnten und erfahren haben, wie die Verbindungs- und Zertifikatsassets verwendet werden, lassen Sie uns das Runbook ausführen und die Ausgabe betrachten.

Jetzt testen

Führen Sie die folgenden Schritte aus, um das Runbook in Aktion zu erleben:

1. Schließen Sie das Fenster, das den Inhalt des Runbooks anzeigt, und kehren Sie zur Übersicht von AzureAutomationScriptTutorial zurück.

2 Wählen Sie „Start“ am oberen Rand des Runbook-Fensters.

3 Bestätigen Sie, dass Sie das Runbook starten möchten, und warten Sie dann einige Sekunden, bis die Ausführung des Runbooks beginnt.

4 Wählen Sie „Output“ (Ausgabe), wie in Abbildung 18.4 gezeigt. Beobachten Sie dann das Konsolenfenster, wenn sich das Runbook in Azure anmeldet, eine Liste von Ressourcengruppen abruft und die Liste der Ressourcen in jeder dieser Gruppen durchläuft und ausgibt.

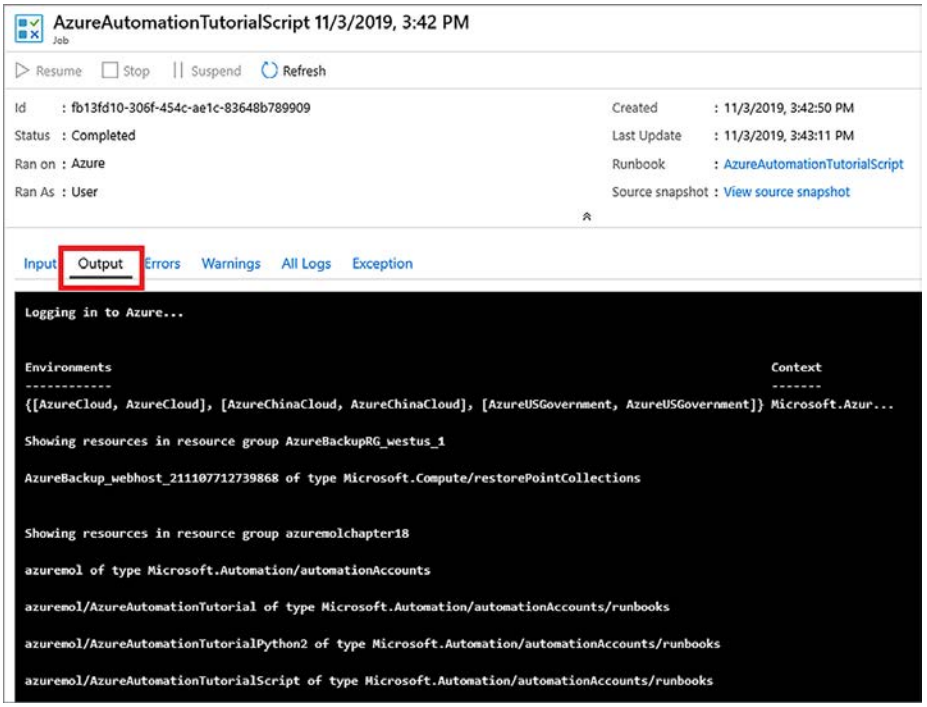


Abbildung 18.4 Die Ausgabe des Runbooks kann angezeigt werden, ebenso wie die erstellten Protokolle oder Fehler und Warnungen. Dieses einfache Beispiel ist nach wenigen Sekunden fertig, komplexere Runbooks können jedoch mehr Zeit beanspruchen. Sie können den Status dieser längeren Runbooks überwachen und bei Bedarf deren Ausführung unterbrechen oder anhalten.

Automation-Runbooks müssen nicht isoliert sein. Ein Runbook kann ein weiteres Runbook ausführen. Mit dieser Möglichkeit können Sie eine komplexe, mehrstufige Automatisierung erstellen und die Duplizierung von Code minimieren. Versuchen Sie bei der Entwicklung von Runbooks, diese in kleine, diskrete Codeblöcke zu zerlegen. Gängige wiederverwendbare Funktionen wie z. B. das Anmelden bei Azure und das Erstellen einer Liste von Ressourcen oder einer Liste von VMs sollten als kleine Runbooks erstellt werden, die in größere Runbooks integriert werden können. Wenn neue PowerShell-Cmdlets veröffentlicht oder Parameter geändert werden, können Sie ein einzelnes freigegebenes Runbook, das diese Cmdlets enthält, schnell aktualisieren, anstatt mehrere unterschiedliche Runbooks aktualisieren zu müssen. Auf den ersten Blick mag es nicht so aussehen, als ob kleinere, wiederverwendbare Runbooks ein wenig zusätzliche Arbeit wert sind, aber wenn Ihre Umgebung und die Verwendung von Automation wächst, werden Sie mir danken! Vieles von dem, was Sie in diesem Buch

gemacht haben, betraf kleinere Bereitstellungen. Denken Sie aber auch einmal darüber nach, wie Sie Anwendungen in großem Umfang bereitstellen und verwalten können.

18.3 PowerShell Desired State Configuration (DSC)

In Kapitel 12 wurde das Konzept der VM-Erweiterungen vorgestellt. Bei einer *Erweiterung* handelt es sich um eine kleine Softwarekomponente, die in einer VM installiert ist, um eine bestimmte Aufgabe zu erfüllen. Die VM-Diagnoseerweiterung wurde auf einer VM installiert, um Leistungskennzahlen und Diagnoseprotokolle von innerhalb der VM an die Azure-Plattform zurückmelden zu können. Das ist großartig, aber wir haben auch ein wenig über die automatische Installation der Software gesprochen.

Sie können PowerShell Desired State Configuration (DSC) als eine Möglichkeit zur Softwareinstallation und Serverkonfiguration verwenden. Mit DSC definieren Sie den gewünschten Zustand der Server-Konfiguration. Sie können z. B. zu installierende Pakete, zu konfigurierende Features oder zu erstellende Dateien definieren. Das Beste an DSC ist, dass es nicht nur für die Erstinstallations- und Konfigurationsaktion angewendet werden kann. Im Laufe der Zeit durchlaufen Server oft Wartungs- oder Fehlerbehebungsereignisse, bei denen Konfigurationen und Pakete manuell geändert werden. Der Server würde dann von dem von Ihnen ursprünglich definierten, gewünschten Zustand abweichen. Abbildung 18.5 zeigt, wie Azure Automation als zentraler Server fungieren kann, der die DSC-Definitionen speichert, damit Ziel-Server ihre Konfigurationen empfangen und ihre Konformität zurückmelden können.

Der Local Configuration Manager (LCM) auf jedem Ziel-Server steuert den Verbindungsprozess mit dem Pull Server von Azure Automation, den Empfang und das Parsen der DSC-Definition sowie die Anwendung und Berichterstattung zur Compliance. Die LCM-Engine kann ohne Pullserver arbeiten, bei dem Sie den Prozess lokal aufrufen, um eine DSC-Definition zu lesen und anzuwenden.

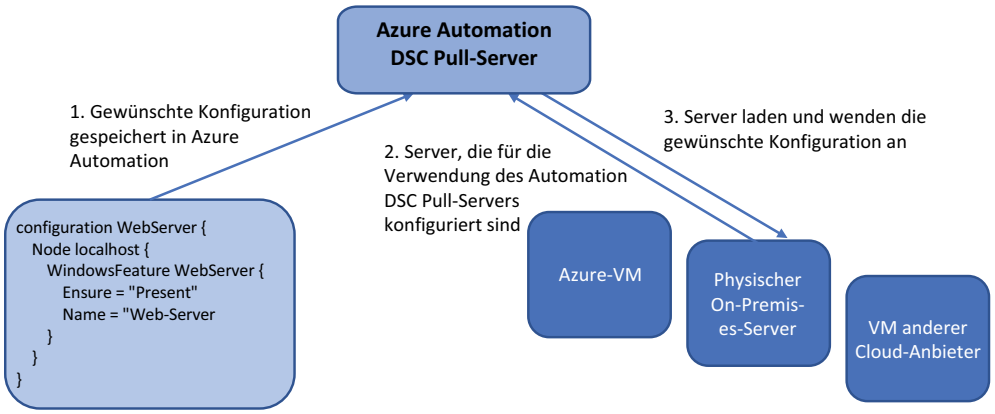


Abbildung 18.5 Die gewünschte Zustandskonfiguration für einen Server wird in Azure Automation erstellt und gespeichert. Das Automation-Konto fungiert als Pullserver, mit dem angeschlossene Server die gewünschte Konfiguration von einem zentralen Ort aus entnehmen können. Für das Korrekturverhalten des Servers können verschiedene Konfigurationsmodi eingestellt werden, wenn dessen Konfiguration vom gewünschten Zustand abweicht.

In diesem Modus übertragen Sie die Konfiguration manuell auf die LCM-Engine und verpassen damit viele der zentralen und oft erforderlichen Kontrollen und Berichte, wenn Sie viele Server verwalten.

Sie haben damit auch Flexibilität in der Art und Weise, wie die Ziel-Server die DSC-Definitionen verarbeiten, die vom Azure Automation Pullserver erhalten wurden. Sie können DSC so konfigurieren, dass es in einem von drei Konfigurationsmodi arbeitet:

- *Nur anwenden* – Ihr gewünschter Zustand wird gepusht und auf den Ziel-Server übertragen. Das war's. Dies entspricht dem Verhalten der Azure Custom Script Extension, da bei der ersten Bereitstellung alle Konfigurationen oder Installationen angewendet werden. Es gibt aber keine Prozesse, die verhindern, dass diese Konfigurationen über den Lebenszyklus des Servers manuell geändert werden.
- *Anwenden und überwachen* – nachdem der Server den gewünschten Zustand erreicht hat, überwacht DSC weiterhin alle Änderungen, die dazu führen, dass der Server von dieser ursprünglichen Konfiguration abweicht. Ein zentraler Bericht kann verwendet werden, um Server anzuzeigen, die nicht mehr dem gewünschten Zustand entsprechen. Diese Konfiguration bietet einen guten Kompromiss zwischen der Notwendigkeit, die Serverkonformität des gewünschten Zustands zu gewährleisten, und der Bereitstellung eines Elements menschlicher Interaktion, um über Korrekturoptionen zu entscheiden.
- *Anwenden und automatisch korrigieren* – die automatisierte und in sich geschlossene Konfiguration wendet den gewünschten Zustand an. Danach überwacht sie alle Abweichungen und korrigiert den Server bei Änderungen automatisch, um sicherzustellen, dass er weiterhin konform ist. Es besteht die Gefahr, dass berechtigte manuelle Änderungen überschrieben und stattdessen in den konfigurierten, gewünschten Zustand zurückgesetzt werden. Dieser Konfigurationsmodus stellt allerdings sicher, dass die von Ihnen zugewiesenen Einstellungen immer Priorität haben.

PowerShell DSC kann in VMs verwendet werden, die bei anderen Cloud-Anbietern sowie auf On-Premises-VMs und physischen Servern ausgeführt werden. Dank .NET Core kann PowerShell DSC auch auf Linux-Servern verwendet werden, sodass es sich nicht um eine reine Windows-Lösung handelt. Dieser herstellerübergreifender Multi-OS-Support macht PowerShell zu einer leistungsstarken Option zur Konfiguration und Verwaltung von Servern in großem Maßstab.

Sie können Ihren eigenen DSC-Pullserver erstellen und warten, aber die integrierten Funktionen von Azure Automation bieten einige zusätzliche Vorteile:

- Die Anmeldeinformationen werden zentral verwaltet, und Zertifikate werden automatisch generiert.
- Die Kommunikation zwischen dem DSC-Pullserver und den Ziel-Servern erfolgt in verschlüsselter Form.
- Integrierte Berichte werden für die DSC-Konformität bereitgestellt und sind mit Log Analytics integriert, um detailliertere Berichte und Warnungen zu generieren.

Dieser Abschnitt ist in hohem Maße ein Crashkurs in PowerShell DSC. Es handelt sich hierbei um eine leistungsstarke Komponente an sich, die seit einigen Jahren weit verbreitet ist. In Kombination mit Azure Automation ist DSC eine gute Wahl für die Automatisierung der Installation und Konfiguration von Software. Denken Sie beispielsweise an die früheren Kapitel über die VM-Skalierungsgruppen. Sie können eine DSC-Konfiguration auf die Skalierungsgruppe mit Azure Automation anwenden. Wenn dann jede VM in der Skalierungsgruppe erstellt wird, wird sie automatisch mit den erforderlichen Komponenten und Anwendungsdateien konfiguriert.

18.3.1 Definieren und Nutzen von PowerShell DSC und einem Azure Automation Pull Server

Ich hoffe, dass Ihnen diese rasante Einführung in PowerShell DSC eine Vorstellung davon vermittelt hat, was möglich ist! Lassen Sie uns PowerShell DSC verwenden, um das vorherige Beispiel der Installation eines einfachen Webservers auf einer VM zu automatisieren.

Jetzt testen

Führen Sie die folgenden Schritte aus, um PowerShell DSC in Aktion zu erleben:

1. Erstellen Sie eine Windows Server 2019 Datacenter VM, und öffnen Sie den TCP-Port 80 für den HTTP-Verkehr. Jetzt geht es richtig los, denn wir sind in Kapitel 18 angekommen! Sie können die VM ganz nach Ihrem Wünsch in der Cloud Shell oder im Azure Portal erstellen. Verwenden Sie die Ressourcengruppe, die Sie in den vorherigen Übungen erstellt haben, z. B. `azuremolchapter18`. Sie können mit den nächsten Schritten fortfahren, wenn die VM bereitgestellt wird.
2. Erstellen Sie auf Ihrem lokalen Rechner eine Datei namens webserver.ps1 und geben Sie den folgenden Code ein. Speichern und schließen Sie die Datei, wenn Sie fertig sind:

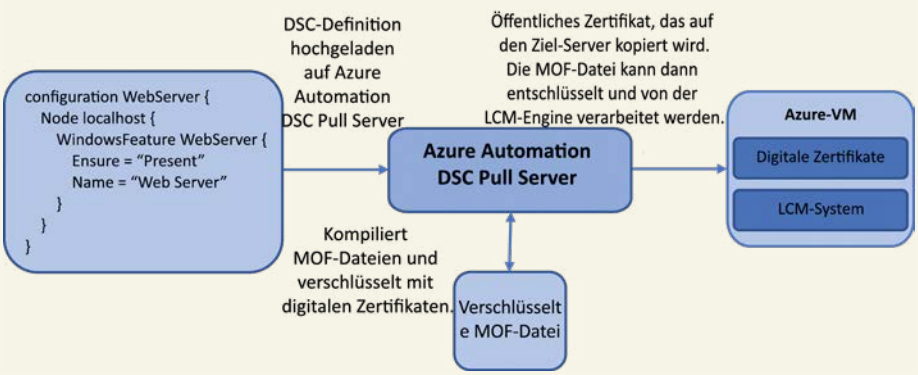
```
configuration WebServer {
    Node localhost {
        WindowsFeature WebServer {
            Ensure = "Present"
            Name = "Web-Server"
        }
    }
}
```

3. Wählen Sie im Azure Portal Ihre Ressourcengruppe aus, und wählen Sie dann Ihr Automation-Konto.
4. Wählen Sie links State Configuration (DSC), wählen Sie die Registerkarte „Configurations“ (Konfigurationen) und wählen Sie oben im Fenster die Option „Add a configuration“ (Konfiguration hinzufügen).
5. Suchen Sie nach der Datei webserver.ps1, und wählen Sie diese aus. Der Konfigurationsname muss mit dem Dateinamen übereinstimmen, also lassen Sie den Standardnamen des Webservers unverändert, und wählen Sie dann OK.

   Es dauert einige Augenblicke, bis die Konfiguration hochgeladen und erstellt ist.
6. Wenn es fertig ist, wählen Sie die Konfiguration aus der Liste aus, und wählen Sie dann „Compile“ (Kompilieren).

Hinter den Kulissen von DSC

Lassen Sie uns eine Pause einlegen und darüber sprechen, was passiert, wenn Sie die Konfiguration kompilieren, wie in der Abbildung in der Seitenleiste dargestellt. Um die DSC-Definitionen zu verteilen, werden Ihre PowerShell-Dateien in eine MOF-Datei (Managed Object Format) umgewandelt. Dieser Dateityp wird nicht nur für PowerShell DSC verwendet und ermöglicht Konfigurationsänderungen an Windows-Komponenten in einer zentralen und gut verständlichen Form. Jede DSC-Definition, nicht nur in Azure Automation, muss kompiliert werden, bevor sie auf einen Zielserver angewendet werden kann. Die LCM-Engine akzeptiert und verarbeitet nur MOF-Dateien.



Der Azure Automation DSC-Pullserver kompiliert automatisch die von Ihnen bereitgestellte DSC-Definition in eine Managed Object Format (MOF)-Datei. Digitale Zertifikate, die von Automation verwaltet werden, werden zur Verschlüsselung der MOF-Datei verwendet. DSC-Zielserver erhalten die erforderlichen öffentlichen digitalen Zertifikate und ermöglichen es der LCM-Engine, die MOF-Datei zu entschlüsseln und zu verarbeiten. Der gewünschte Zustand kann dann auf den Server angewendet werden.

Da die MOF-Datei den kompletten Zustand Ihrer Server definiert, sollten Sie ihren Inhalt schützen. Wenn ein Angreifer alle installierten Anwendungskomponenten und den Speicherort verschiedener Konfigurationsdateien und benutzerdefinierten Codes kennen würde, würde dies zu einer größeren Gefährdung Ihrer Server erhöhen. Neuere Versionen von PowerShell verschlüsseln die gesamte MOF-Datei. Azure Automation generiert automatisch die erforderlichen digitalen Zertifikate und Schlüssel, wenn ein Zielserver für DSC konfiguriert ist, sodass Sie nahtlos verschlüsselte MOF-Dateien verwenden können. Automation verschlüsselt nicht nur die MOF-Datei, sondern auch den Datenverkehr zwischen DSC-Pullserver und Zielknoten.

Der Kompilierungsprozess in Azure Automation konvertiert die von Ihnen bereitgestellte DSC-Definition in eine MOF-Datei und verschlüsselt auch die MOF-Datei mit den digitalen Zertifikaten und Schlüsseln. Der Prozess der Kompilierung Ihrer DSC-Definition dauert nur wenige Sekunden, sichert aber Ihre Umgebung in hohem Maße – nur ein weiteres Beispiel dafür, dass Azure Ihre Ressourcen standardmäßig sichert!

7. Um die Konfiguration auf Ihre VM anzuwenden, wählen Sie die Registerkarte „Nodes“ (Knoten) im Fenster „State configuration (DSC)“ (Statuskonfiguration (DSC)). Wählen Sie „Add“ (Hinzufügen) und dann die VM, die Sie in den vorherigen Schritten erstellt haben.
8. Wählen Sie „Connect“ (Verbinden).
9. Wählen Sie im Dropdown-Menü „Configuration Name“ (Name der Knotenkonfiguration) die Option webserver.localhost aus.
10. Legen Sie den Konfigurationsmodus auf ApplyAndMonitor fest, und wählen Sie dann OK.

    Es dauert ein oder zwei Minuten, bis die VM den Azure PowerShell DSC Pull Server verwenden und den anfänglich gewünschten Zustand anwenden kann.
11. Wenn das Azure Portal meldet, dass die Konfiguration angewendet wurde, wählen Sie Ihre Ressourcengruppe und dann die VM aus, die Sie in den vorherigen Schritten erstellt haben.

12 Haben Sie den TCP-Port 80 für die VM geöffnet, als Sie die VM erstellt haben? Wenn nicht, erstellen Sie eine Gruppenregeln für die Netzwerksicherheit, um den Datenverkehr zuzulassen, und öffnen Sie dann die öffentliche IP-Adresse der VM in einem Webbrowser. Der DSC-Prozess installierte den IIS-Webserver und die Standard-Webseitenlasten, wie in Abbildung 18.6 dargestellt.

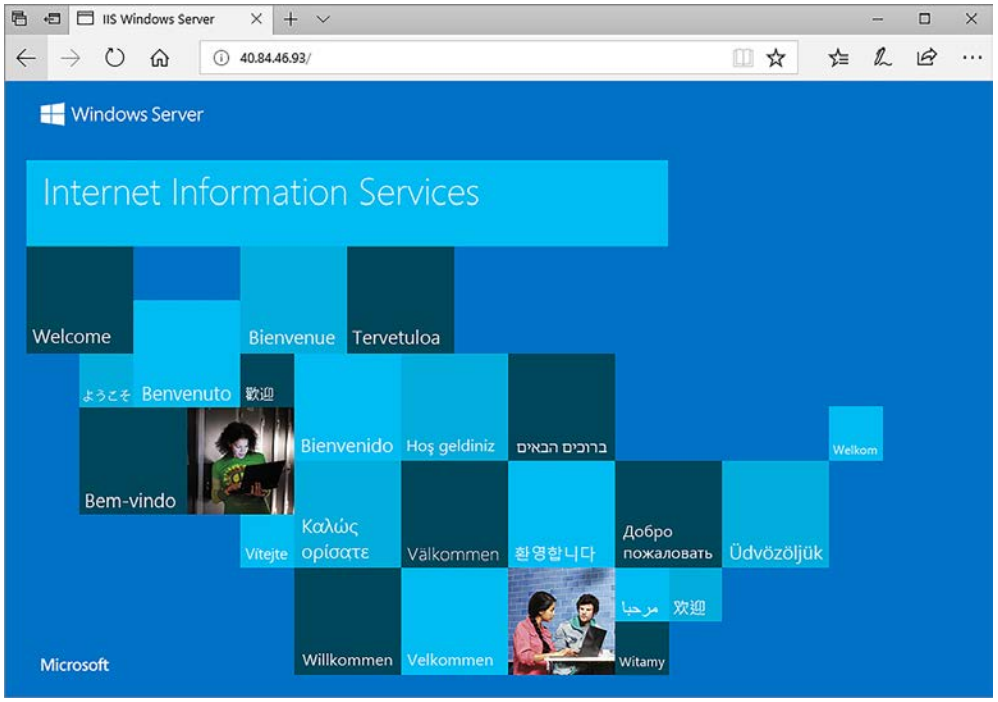


Abbildung 18.6 Nachdem die VM mit Azure Automation DSC verbunden wurde, wird der gewünschte Zustand übernommen und der IIS-Webserver installiert.

Dieses einfache Beispiel von PowerShell DSC installiert nur die Webserver-Funktion. Sie können PowerShell DSC verwenden, um den IIS-Webserver zu konfigurieren oder Ihren Anwendungscode auf die VM zu kopieren und die Website auszuführen. Komplexe DSC-Definitionen können verwendet werden, um die VM darauf vorzubereiten, den Verkehr zu Ihren Pizzeria-Kund*innen ohne manuelle Interaktion zu bedienen. Denken Sie noch einmal darüber nach, wie Sie Ihre Anwendungen gestalten sollten, dass sie automatisch skaliert werden. Die VM kann es kaum erwarten, dass sich jemand anmeldet und alles manuell installiert und konfiguriert!

18.4 Übung: DSC mit Linux

Um nachzuweisen, dass PowerShell DSC auf Linux-Servern funktioniert, erstellen wir eine Ubuntu-VM, installieren die notwendigen Voraussetzungen und installieren dann einen einfachen NGINX-Webserver mit DSC. In der Produktion können Sie ein benutzungsdefiniertes VM-Image mit bereits installierten Verwaltungskomponenten verwenden und dann PowerShell DSC-Definitionen wie gewohnt anwenden:

1. PowerShell DSC für Linux hat ein paar Einschränkungen bei den Linux-Distributionen, die es ohne zusätzliche Konfiguration unterstützt. Um diese Übung am Ende des Kapitels so einfach wie möglich zu halten, erstellen Sie eine CentOS 7.7 oder höher VM, und öffnen Sie Port 80.
2. Wählen Sie in Ihrem Azure Automation-Konto „Modules“ (Module) aus dem Menü auf der linken Seite.
3. Wählen Sie „Browse Gallery“ (Galerie durchsuchen), suchen und importieren Sie dann das Modul nx zur Verwaltung von Linux-DSC-Ressourcen.
4. Erstellen Sie auf Ihrem lokalen Rechner einen Dateinamen httpd.ps1, und geben Sie den folgenden Code ein:

```
configuration httpd {
    Import-DSCResource -Module nx
    Node localhost {
        nxPackage httpd {
            Name = "httpd"
            Ensure = "Present"
            PackageManager = "yum"
        }
        nxService httpd {
            Name = "httpd"
            State = "running"
            Enabled = $true
            Controller = "systemd"
        }
    }
}
```

5. Fügen Sie dem Azure Automation-Konto eine DSC-Konfiguration hinzu, laden Sie die httpd.ps1-Datei hoch, und kompilieren Sie die Konfiguration.
6. Fügen Sie einen DSC-Knoten zu Ihrem Azure Automation-Konto hinzu, wählen Sie Ihre CentOS-VM aus, und wählen Sie dann Ihren httpd.localhost-Knoten-Konfigurationsnamen.

   Es dauert wieder ein oder zwei Minuten, bis die VM die gewünschte Konfiguration angewendet hat. Sie können die Liste der verbundenen VMs und deren Konformitätsstatus im Fenster der DSC-Knoten einsehen. Die VM meldet konform als Status, wenn der LCM die MOF-Datei akzeptiert und angewendet hat. Die Befehle zur Installation und Konfiguration der erforderlichen httpd-Pakete innerhalb der VM können jedoch noch ein oder zwei Minuten benötigen.
7. Wählen Sie Ihre CentOS-VM im Azure Portal aus, rufen Sie ihre öffentliche IP-Adresse ab, und geben Sie die IP-Adresse Ihrer VM in einem Webbrowser ein, um den von DSC installierten Webserver anzuzeigen. Wird die Website nicht geladen, warten Sie ein oder zwei Minuten, bis der Installationsprozess abgeschlossen ist, und aktualisieren Sie danach die Seite.

Wenn Sie die schöne neue Welt von Microsoft und Linux wirklich erleben möchten, können Sie PowerShell auf Ihrer Linux-VM installieren. Führen Sie die Schritte zur schnellen Einrichtung unter http://mng.bz/VgyP durch, um zu verstehen, wie plattformübergreifende PowerShell-Skripte heutzutage funktionieren können!

19 Azure Container

Container, Docker und Kubernetes haben in wenigen Jahren sehr viele Fans gefunden. Ähnlich wie Server-Virtualisierung Mitte der 2010er Jahre die Art und Weise verändert hat, wie IT-Abteilungen ihre Rechenzentren betreiben, revolutionieren moderne Container-Tools und Orchestratoren nun die Art und Weise, wie wir Anwendungen erstellen und ausführen. Beim zunehmenden Container-Einsatz handelt es sich um eine vom Cloud Computing unabhängige Tendenz, zusammen bieten sie allerdings eine großartige Möglichkeit, Anwendungen mit einem cloud-nativen Ansatz zu entwickeln. Ganze Bücher wurden zum Thema Docker und Kubernetes verfasst, doch lassen Sie uns zunächst bei einer intensiven Einführung darauf eingehen, wie Sie Container in Azure schnell betreiben können. Für Container gibt es eine leistungsstarke Suite von Azure-Diensten, die sich weitgehend am PaaS-Ansatz orientiert. Sie können sich auf die Erstellung und Ausführung Ihrer Anwendungen konzentrieren, anstatt Containerinfrastruktur, Orchestrierung und Clusterkomponenten zu verwalten.

In diesem Kapitel untersuchen wir, was Container sind, wie Docker darin eingebunden ist und wie Sie von Kubernetes profitieren können. Um zu erfahren, wie Sie schnell entweder eine einzelne Container-Instanz oder mehrere Container-Instanzen in einem Cluster ausführen können, untersuchen wir Azure Container Instances (ACI) und Azure Kubernetes Service (AKS).

19.1 Was sind Container?

In den letzten Jahren nahmen das Interesse und die Akzeptanz rund um Container dramatisch zu. Ich wäre sehr überrascht, wenn Sie nicht wenigstens von einem Unternehmen gehört hätten, das Docker als Trend ignoriert hätte. Aber was genau ist ein Container, und was hat Docker damit zu tun?

Zuerst erläutern wir einen traditionellen Virtualisierungshost, der VMs ausführt. Abbildung 19.1 ähnelt dem Diagramm aus dem Kapitel 1, in dem jede VM über ihre eigene virtuelle Hardware und ihr eigenes Gastbetriebssystem verfügt.

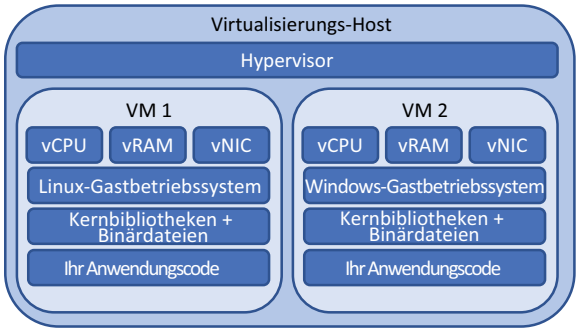


Abbildung 19.1 Bei einer traditionellen VM-Infrastruktur bietet der Hypervisor auf jedem Virtualisierungshost eine Isolationsschicht, indem er jeder VM einen eigenen Satz virtueller Hardwaregeräte zur Verfügung stellt, wie beispielsweise eine virtuelle CPU, virtuellen RAM und virtuelle NICs. Die VM installiert ein Gastbetriebssystem, wie beispielsweise Ubuntu Linux oder Windows Server, das diese virtuelle Hardware nutzen kann. Schließlich installieren Sie Ihre Anwendung und alle erforderlichen Bibliotheken. Dieser Isolierungsgrad macht VMs sehr sicher, bringt aber eine zusätzliche Overhead-Schicht in Bezug auf Rechenressourcen, Speicher und Startzeiten mit sich.

Ein Container macht die virtuelle Hardware und das Gastbetriebssystem überflüssig. Alles, was in einem Container enthalten ist, sind die zum Ausführen Ihrer App erforderlichen Kernanwendungen und Bibliotheken, wie in Abbildung 19.2 dargestellt.

Viele VMs können auf einem zentralen Hypervisor ausgeführt werden. Jeder VM wird dabei ein eigenes virtuelles Gastbetriebssystem, virtuelle Hardware und Anwendungsstapel zugewiesen. Der Hypervisor verwaltet Anfragen von der virtuellen Hardware jeder VM, plant die Zuweisung und gemeinsame Nutzung dieser physischen Hardwareressourcen und erzwingt die Sicherheit und Isolation jeder VM. Die Arbeit des Hypervisors ist in Abbildung 19.3 dargestellt.

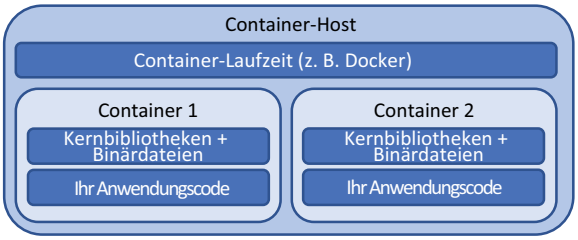


Abbildung 19.2 Ein Container enthält nur die Kernbibliotheken, Binärdateien und Anwendungscodes, die zum Ausführen einer App erforderlich sind. Der Container ist leicht und portabel, da er das Gastbetriebssystem und die virtuelle Hardware-Ebene beseitigt und damit auch die Größe des Containers auf dem Datenträger und die Startzeiten reduziert.

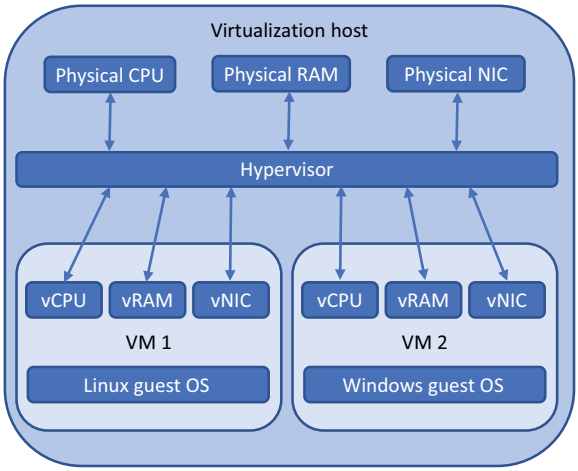


Abbildung 19.3 In einem traditionellen VM-Host ermöglicht der Hypervisor die Planung von Anforderungen der virtuellen Hardware in jeder VM an die zugrunde liegende physische Hardware und Infrastruktur. Der Hypervisor ist sich in der Regel nicht bewusst, welche spezifischen Anweisungen das Gastbetriebssystem für die physische CPU-Zeit plant. Er weiß lediglich, dass CPU-Zeit benötigt wird.

Mehrere Container können auch auf einem einzigen Host ausgeführt werden. Der Container-Host empfängt die verschiedenen Systemaufrufe von jedem Container und plant die Zuordnung und Verteilung dieser Anforderungen über einen gemeinsamen Basiskern, Betriebssystem- und Hardwareressourcen hinweg. Container bieten eine logische Trennung von Anwendungsprozessen. Die Funktionsweise der Container-Laufzeit ist in Abbildung 19.4 dargestellt.

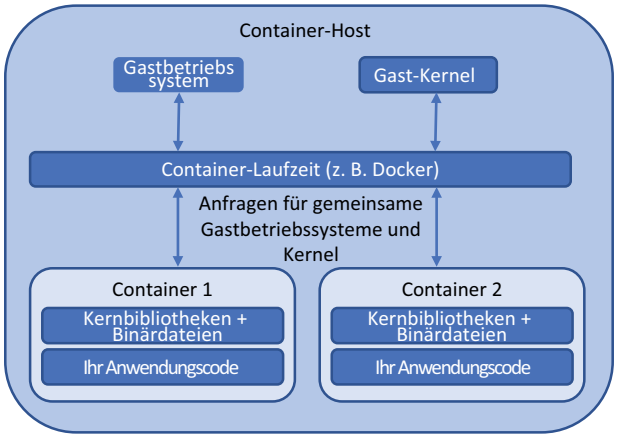


Abbildung 19.4 Container teilen sich ein gemeinsames Gastbetriebssystem und einen gemeinsamen Kernel. Die Container-Laufzeit verarbeitet die Anfragen der Container an den gemeinsamen Kernel. Jeder Container wird in einem isolierten Benutzerbereich ausgeführt. Einige zusätzliche Sicherheitsmerkmale schützen dabei die Container voreinander.

Container verfügen in der Regel über eine viel schlankere Struktur als VMs. Container können schneller als VMs, oft innerhalb von Sekunden statt Minuten, gestartet werden. Die Größe eines Container-Images beträgt typischerweise nur Dutzende oder Hunderte von MBs, im Vergleich zu Dutzenden von GBs bei VMs. Sicherheitseinschränkungen und Kontrollen sind auch hier vorhanden, doch Sie sollten daran denken, dass jeder Container technisch gesehen mit anderen Containern denselben Host teilt.

Jetzt testen

Es kann einige Minuten in Anspruch nehmen, um einen AKS Cluster für die kommenden Übungen zu erstellen. Führen Sie deshalb bitte die folgenden Schritte aus, und fahren Sie dann mit der Lektüre des Kapitels fort.

1. Öffnen Sie das Azure Portal, und wählen Sie das Cloud Shell-Symbol aus dem oberen Menü.
2. Erstellen Sie eine Ressourcengruppe. Geben Sie einen Namen, z. B. `azuremolchapter19` an, und einen Standort, z. B. `eastus`. Die Verfügbarkeit von AKS kann je nach Region variieren, wählen Sie deshalb eine größere Region wie z. B. `eastus` oder `westeurope` aus. (Eine aktuelle Liste der verfügbaren Regionen finden Sie unter https://azure.microsoft.com/regions/services.)

```
az group create --name azuremolchapter19 --location eastus
```

3 Um einen Kubernetes-Cluster zu erstellen, geben Sie `--node-count` als `2` an, und verwenden Sie Skalierungsgruppen und Verfügbarkeitszonen (was Sie in früheren Kapiteln gelernt haben):

```
az aks create \
  --resource-group azuremolchapter19 \
  --name azuremol \
  --node-count 2 \
  --vm-set-type VirtualMachineScaleSets \
  --zones 1 2 3 \
  --no-wait
```

Der letzte Parameter `--no-wait` gibt die Kontrolle an Cloud Shell zurück, während der Rest des Clusters erstellt wird. Sie können weiterlesen, während der Cluster bereitgestellt wird.

Docker beteiligt sich an der Container-Bewegung mit einer Reihe von Tools und Standardformaten, die definieren, wie ein Container erstellt und ausgeführt werden kann. Docker nutzt bestehende Funktionen auf Linux- und Windows-Kernel-Ebene und bietet eine portable, konsistente Container-Erfahrung auf allen Plattformen. Ein*e Entwickler*in kann auf seinem macOS-Laptop einen Docker-Container erstellen, seine App validieren und testen und dann den genauen Docker-Container unverändert in einem traditionelleren Linux- oder Windows-basierten Server-Cluster On-Premises oder auf Azure ausführen. Alle erforderlichen Anwendungsbinärdateien, Bibliotheken und Konfigurationsdateien werden als Containerteil gebündelt, sodass das zugrunde liegende Host-Betriebssystem keine Auswirkungen auf das Design hat und sie auch nicht einschränkt.

Die Bedeutung von Docker darf hier nicht übersehen werden. Die Begriffe *Container* und *Docker* werden oft austauschbar verwendet, obwohl es technisch nicht korrekt ist. Bei Docker handelt es sich um eine Reihe von Tools, die den Aufbau und Betrieb von Containern auf eine konsistente, zuverlässige und portable Weise unterstützen. Die einfache Handhabung dieser Tools führte zu einer schnellen Akzeptanz und machte die zugrunde liegende Containertechnologie, die seit über einem Jahrzehnt in der einen oder anderen Form bestand, zu einer Mainstream-Technologie. Die aktive Nutzung von Containern und der Docker-Plattform durch Entwickler*innen zwang die IT-Abteilungen dazu, ihren Rückstand aufzuholen.

Docker beteiligt sich an der Open Container Initiative. Das Format und die Spezifikationen, die Docker für die Verpackung und den Betrieb eines Containers definiert hat, bildeten einige der Grundprinzipien für dieses Projekt. Die Arbeit des Dockers wurde fortgeführt und von anderen weiter ausgebaut. Zu den wichtigsten Mitwirkenden in der Container-Bewegung gehören IBM und Red Hat, die einige der Kerndesigns und den Code für den Betrieb der aktuellen Container-Plattformen beisteuern. Die Open Container Initiative und das Designformat für Container-Packaging und -laufzeiten spielt eine wichtige Rolle und ermöglicht jedem Anbieter, seine eigenen Tools für die gängigen Formate zu nutzen. Auf diese Weise können Sie den zugrunde liegenden Container zwischen den Plattformen verschieben und die Einheitlichkeit der Kerneigenschaften wird gewährleistet.

19.2 Das Microservice-Konzept für Anwendungen

Wenn Container ein Konzept der Isolierung ähnlich wie VMs bieten, können Sie dann die gleiche Art von Workloads wie in einer VM ausführen? Nun, die Antwort lautet hier ja und nein. Nur weil Sie etwas tun können, heißt das nicht unbedingt, dass Sie es sollten! Container können verwendet werden, um beliebige Workloads auszuführen,

mit denen Sie vertraut sind. Sie bieten auch Vorteile in Bezug auf Portabilität und Orchestrierungsfunktionen, auf die wir später in Abschnitt 19.4 eingehen. Um die Vorteile von Containern zu maximieren und die größtmöglichen Vorteile daraus zu ziehen, sollten Sie ein etwas anderes mentales Modell bei Ihrer Arbeit mit Containern nutzen. In Abbildung 19.5 wird das traditionelle Anwendungsmodell mit einem Microservice-Ansatz verglichen.

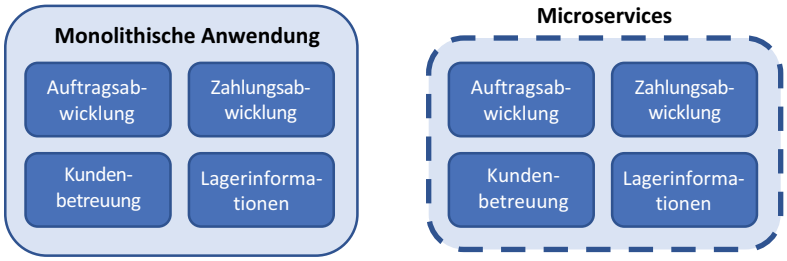


Abbildung 19.5 In einer traditionellen monolithischen Struktur wird die gesamte Anwendung als eine einzige Anwendung ausgeführt. Die Anwendung kann verschiedene Komponenten enthalten, aber sie wird von einer einzigen Installation ausgeführt und als eine zentrale Instanz gepatcht und aktualisiert. Bei Microservices wird jede Komponente in einen eigenen Anwendungsdienst und eine Ausführungseinheit unterteilt. Jede Komponente kann unabhängig von anderen aktualisiert, gepatcht und skaliert werden.

Eine Standard-VM beinhaltet eine vollständige Installation des Gastbetriebssystems, wie z. B. Ubuntu oder Windows Server. Diese Basisinstallation des Betriebssystems umfasst Hunderte von Komponenten, Bibliotheken und Tools. Anschließend installieren Sie weitere Bibliotheken und Anwendungen, unter anderem für den NGINX-Webserver oder Microsoft SQL Server. Stellen Sie abschließend Ihren Anwendungscode bereit. Diese VM führt typischerweise einen großen Teil, wenn nicht sogar die komplette Anwendung aus. Es handelt sich um eine zentrale Instanz zur Installation und Ausführung von Anwendungen. Um die Leistung zu verbessern, können Sie der VM mehr Arbeitsspeicher oder CPU hinzufügen (vertikale Skalierung, wie in den vorangegangenen Kapiteln beschrieben) oder die Anzahl der Instanzen erhöhen, die Ihre Anwendung ausführen (horizontale Skalierung, wie bei Skalierungssätzen). Das Erstellen mehrerer Anwendungsinstanzen funktioniert nur, wenn Ihre Anwendung clusterfähig ist. Es handelt sich oft um eine Form von gemeinsamem Speicher, um einen konsistenten Zustand für alle Anwendungsinstanzen zu ermöglichen. Diese traditionelle Form der Bereitstellung wird als *monolithische* Anwendung bezeichnet.

Ein anderer Ansatz, wie Sie Anwendungen entwickeln und ausführen können, besteht darin, die Dinge in kleinere, praktische Komponenten aufzuteilen. Es handelt sich dabei um einen *Microservice-Ansatz* für die Entwicklung und Bereitstellung von Anwendungen. Jeder Microservice übernimmt die Verantwortung für einen kleinen Teil der gesamten Anwendungsumgebung. Microservices können unabhängig von der übrigen Anwendungsumgebung wachsen, skaliert und aktualisiert werden.

Dieses Modell kann zunächst eine Herausforderung darstellen, denn Entwicklungs- und IT-Teams müssen zunächst lernen, eine andere Art der Erstellung und Bereitstellung von Anwendungen zu wählen. Allerdings sind Container für den Microservice-Ansatz optimal geeignet. Entwickler*innen können kleinere, inkrementelle Updates schneller bereitstellen, als dies beim monolithischen Ansatz zur Anwendungsentwicklung der Fall ist. Microservices und Container eignen sich auch hervorragend für kontinuierliche Integration und Continuous Delivery (CI/CD)-Workflows, bei denen Sie Updates einfacher erstellen, testen und bereitstellen können. Ihre Kund*innen erhalten neue Funktionen oder Fehlerkorrekturen schneller als sonst, sodass Ihr Unternehmen davon profitieren kann.

Microservices mit Azure Service Fabric

Dieses Kapitel befasst sich mit Docker-Containern und der Orchestrierung mit Kubernetes. Allerdings gibt es einen weiteren Azure-Dienst, der einen ähnlichen Ansatz bei der Anwendungsentwicklung nutzt und das Microservice-Modell vorantreibt. Azure Service Fabric wurde vor einigen Jahren entwickelt. Es handelte sich dabei historisch gesehen um einen Windows-basierten Ansatz zur Erstellung von Anwendungen, bei dem jede Komponente ihrem eigenen Microservice zugeordnet wird. Service Fabric verfolgt den Ausführungsstandort jeder Microservice-Komponente in einem Cluster, damit Dienste sie entdecken und miteinander kommunizieren können. Außerdem verwaltet es Redundanz und Skalierung.

Viele zentrale Azure-Dienste nutzen Service Fabric als technologische Grundlage, einschließlich Cosmos DB. Dies soll Ihnen einen Eindruck davon vermitteln, wie leistungsfähig Service Fabric sein kann! Service Fabric selbst läuft auf der Basis der Skalierungsgruppen von Virtual Machines. Sie kennen das eine oder andere über Skalierungsgruppen, oder?

Die Service Fabric-Plattform ist inzwischen gereift und kann nun sowohl Windows als auch Linux als Gastbetriebssystem verwenden, sodass Sie Ihre App mit der Programmiersprache Ihrer Wahl erstellen können. Hier ist ein weiteres Beispiel für die Wahl in Azure: Sie können flexibel wählen, wie Sie Ihre Container-Anwendungen verwalten und orchestrieren möchten. Sowohl Service Fabric als auch AKS bieten ausgezeichnete Vorteile und Anwendungsfälle.

Service Fabric ist eine hervorragende Wahl und ein guter Ausgangspunkt, wenn Sie derzeit Microservices außerhalb von Containern entwickeln oder entwickeln möchten. Anwendungen, die auf der Grundlage des Akteursmodells entwickelt wurden, eignen sich ebenfalls hervorragend, da Service Fabric ursprünglich mit diesem Programmiermodell im Hinterkopf entwickelt wurde. Service Fabric bietet einen einheitlichen Ansatz zur Verarbeitung sowohl traditioneller Microservice- als auch containerbasierter Anwendungen. Wenn Sie sich für den Einsatz von Containern bei anderen Workloads entscheiden, können Sie dieselben Service Fabric-Verwaltungstools und -Schnittstellen verwenden, um alle Ihre Anwendungsumgebungen zu verwalten.

AKS kann sich als eine bessere Wahl erweisen, um von Anfang an einen containerorientierten Anwendungsansatz zu verfolgen. Das Wachstum und die Einführung von Kubernetes bieten Ihnen dabei ein erstklassiges Container-Erlebnis. Sie können mit Linux- und Windows-Containern in AKS ausgeführt werden.

19.3 Azure Container-Instanzen

Nun haben Sie ein etwas besseres Verständnis darüber, was Container sind und ihre Verwendung. Lassen Sie uns jetzt loslegen und eine Basisinstanz für ein Pizzageschäft erstellen. Hier kommt ein Beispiel aus früheren Kapiteln, in denen Sie eine grundlegende VM erstellt haben, die Ihre Website ausgeführt oder die App in Webanwendungen bereitgestellt hat. In beiden Fällen mussten Sie die VM oder Webanwendung erstellen, sich mit ihr verbinden und dann eine einfache Webseite dafür bereitstellen. Kann die Leistung von Containern den gesamten Prozess um so viel einfacher gestalten? Auf jeden Fall!

Mit einem übersichtlichen Dienst namens Azure Container Instances (ACI) können Sie Container in wenigen Sekunden erstellen und ausführen. Es gibt keine im Voraus zu erstellenden und zu konfigurierenden Netzwerkressourcen, und Sie bezahlen für jede Container-Instanz pro Sekunde. Wenn Sie noch nie Container verwendet haben und nichts lokal auf Ihrem Computer installieren möchten, bietet ACI Ihnen eine gute Möglichkeit, um die Technologie auszuprobieren.

Wir erstellen eine Container-Instanz und zeigen Ihnen, wie Sie einen Pizzeria-Shop schnell in Betrieb nehmen können. Lediglich ein Befehl ist erforderlich, um eine Container-Instanz auszuführen. Abbildung 19.6 zeigt Ihnen, wie Sie viele Komponenten zusammenführen können, um dies im Hintergrund zu realisieren. Wir betrachten die Komponenten eines Dockerfile und Docker Hub, nachdem Sie die Ausführung der Container-Instanz eingeleitet haben.

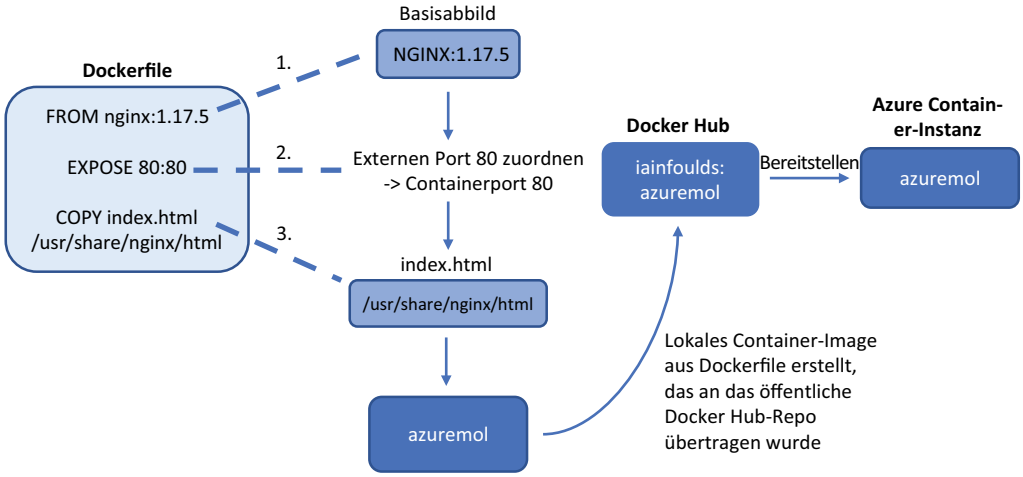


Abbildung 19.6 Ein Dockerfile wurde verwendet, um ein komplettes Container-Image, Azuremol, zu erstellen. Dieses Image wurde an eine öffentliche Online-Registry namens Docker Hub weitergeleitet. Sie können nun eine Container-Instanz mit diesem vordefinierten öffentlichen Image aus dem Docker Hub erstellen, das ein ausführbares Anwendungsimage bereitstellt.

Jetzt testen

Befolgen Sie die folgenden Schritte, um eine Azure-Container-Instanz zu erstellen, die eine Basis-Website ausführt.

1. Öffnen Sie das Azure Portal, und wählen Sie das Cloud Shell-Symbol aus dem oberen Menü.
2. Erstellen Sie eine Container-Instanz, und geben Sie an, dass Sie eine öffentliche IP-Adresse haben und Port 80 öffnen möchten:

```
az container create \
  --resource-group azuremolchapter19 \
  --name azuremol \
  --image iainfoulds/azuremol \
  --ip-address public \
  --ports 80
```

In dieser Übung wird ein Beispiel-Image verwendet, das ich für Sie erstellt habe. Wir werden noch ein wenig genauer darauf eingehen, sobald der Container in Betrieb ist.

3 Schauen Sie sich die Ausgabe des Befehls an, um den Container zu erstellen.

Im Abschnitt „Ereignisse“ sehen Sie, wie das Image aus dem Docker Hub gezogen (heruntergeladen) wird, ein Container erstellt wird und der Container dann gestartet wird.

Einige CPU- und Arbeitsspeicher-Reservierungen sind ebenfalls zugeordnet und können bei Bedarf angepasst werden. Eine öffentliche IP-Adresse wird zusammen mit einigen Informationen zum Container angezeigt, wie z. B. dem Bereitstellungsstatus, dem Betriebssystemtyp und der Neustartrichtlinie.

4 Sie können nur nach der zugewiesenen öffentlichen IP-Adresse suchen, um die Basis-Website zu öffnen, die im Container ausgeführt wird:

```
az container show \
  --resource-group azuremolchapter19 \
  --name azuremol \
  --query ipAddress.ip \
  --output tsv
```

5 Öffnen Sie die öffentliche IP-Adresse Ihrer Container-Instanz in einem Webbrowser. Das grundlegende Modell des Pizzageschäfts sollte angezeigt werden, wie in Abbildung 19.7 dargestellt.

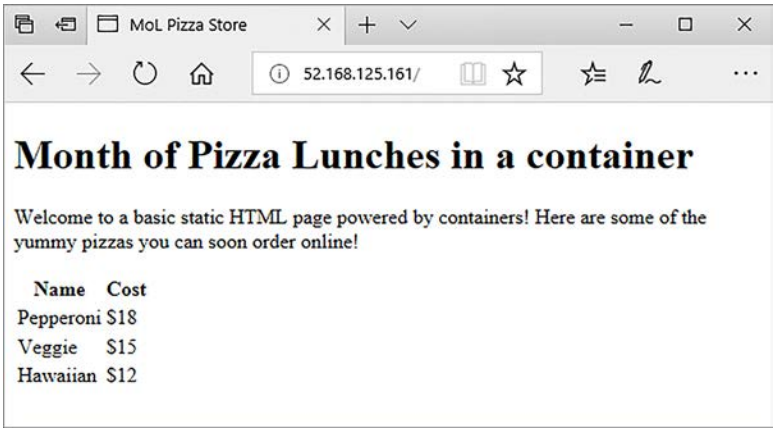


Abbildung 19.7 Wenn Sie eine Container-Instanz erstellen, läuft die Website des Pizzageschäfts ohne zusätzliche Konfiguration. Alle Konfigurationen und Inhalte sind im Container-Image enthalten. Diese schnelle Übung verdeutlicht die Portabilität und Leistungsfähigkeit von Containern. Nach der Erstellung eines Container-Image ist Ihre App einsatzbereit, sobald eine neue Container-Instanz bereitgestellt wird.

Betrachten wir das Container-Image. Ich möchte nicht zu weit auf die Einzelheiten von Docker und die Erstellung von Container-Images eingehen. Sie sollten an dieser Stelle vor allem verstehen, woher dieses Image stammt und wie es die Website ohne zusätzliche Konfiguration ausführt.

Das Image wird aus einer Konfigurationsdefinition mit dem Namen *Dockerfile* erstellt. In einem Dockerfile definieren Sie die Basisplattform, jede Konfiguration, die Sie anwenden möchten, und alle Befehle zum Ausführen oder Kopieren von Dateien.

Dockerfiles sind oft komplexer als das folgende Beispiel, das zum Aufbau des Azuremol-Beispiel-Containers verwendet wurde:

```
FROM nginx:1.17.5

EXPOSE 80:80

COPY index.html /usr/share/nginx/html
```

Bei Verwendung dieses Dockerfile zur Erstellung eines Docker-Container-Images wurde NGINX als Quell-Image verwendet und die Beispiel-Webseite wurde darein kopiert. Dieser Container wurde dann an den Docker Hub weitergeleitet, ein öffentliches Online-Repository, das Docker für die Freigabe und Bereitstellung von Containern bereitstellt. Um die Container-Instanz bereitzustellen, haben Sie iainfoulds/azuremol als Container-Image angegeben. Azure sah im Docker Hub nach und fand ein Repository namens iainfoulds und darin ein Image namens Azuremol.

Betrachten wir jede Zeile von Dockerfile:

- `FROM nginx:1.17.5` In den vorherigen Kapiteln haben Sie eine grundlegende VM erstellt. Dann haben Sie darüber mit einem SSH eine Verbindung hergestellt und dann den NGINX-Webserver manuell installiert. Im Beispiel-Dockerfile wird dies alles in einer Zeile erledigt. Diese Zeile besagt, dass der Container auf einem vorhandenen Container-Image basiert, das mit NGINX vorinstalliert ist. Die `1.17.5` ist die Version des öffentlichen NGINX-Container-Images, das verwendet werden soll. Zum Zeitpunkt der Erstellung ist dies die neueste Version. Es ist empfehlenswert, eine bestimmte Versionsnummer anzugeben. Wenn Sie keine Versionsnummer vergeben, wird immer die neueste Version verwendet. In der Theorie hört sich das gut an, aber Microservices-Anwendungen können auf eine ziemlich große Anzahl aktiver Container skaliert werden. Um also sicherzustellen, dass Sie eine konsistente Umgebung haben, müssen Sie die genaue Versionsnummer jeder verwendeten Komponente kontrollieren.
- `EXPOSE 80:80` – um den Zugriff auf Ihre VM aus den vorangegangenen Kapiteln zu ermöglichen, haben Sie eine NSG-Regel erstellt, die Port 80 zulässt. Im Dockerfile weist diese Zeile den Container an, Port 80 zu öffnen und dem internen Port 80 zuzuordnen. Als Sie Ihre Container-Instanz mit `az container create` erstellt haben, haben Sie auch angegeben, dass die Azure-Plattform den Traffic mit `--ports 80` zulassen soll. Das sind alle Abläufe im virtuellen Netzwerk, die Sie berücksichtigen müssen!
- `COPY Index.html/usr/share/nginx/html` – der letzte Teil besteht darin, Ihre Anwendung in den Container zu bringen. In den vorhergehenden Kapiteln haben Sie Git verwendet, um die Webseite des Beispiel-Pizzaladens zu erstellen und diese dann auf Ihre Web App zu übertragen. Mit dem Dockerfile `COPY` Sie die Datei index.html in das lokale Verzeichnis /usr/share/nginx/html im Container. So einfach ist das.

Für Ihre eigenen Szenarien können Sie einen Dockerfile definieren, das ein anderes Basis-Image wie z. B. Node.js oder Python verwendet. Sie installieren dann alle zusätzlichen unterstützenden und erforderlichen Bibliotheken oder Pakete, ziehen Ihren Anwendungscode aus der Quellenkontrolle, wie GitHub, und stellen Ihre Anwendung bereit. Dieser Dockerfile wird verwendet, um Container-Images zu erstellen, die dann in einem privaten Container-Verzeichnis gespeichert werden. Sie werden nicht in einem öffentlichen Docker Hub-Repository gespeichert, wie dies im Beispiel gezeigt wurde.

Azure Container Registry

Sie denken vielleicht, dass Docker Hub einfach großartig klingt, und fragen sich, ob Azure auch eine derart großartige Option bietet? Das tut es! Sie müssen an dieser Stelle ein Dockerfile und ein Container-Image erstellen. Es handelt sich hierbei leider um keine zweiminütige Übung, und es gibt viel zu beachten in diesem Kapitel. Sie können Azure Container Registry (ACR) und AKS einfach integrieren, sodass die beiden Dienste gut zusammenarbeiten. Sie können jedoch Ihre eigenen Images aus einem Dockerfile in der Cloud Shell erstellen. Ich würde Ihnen unbedingt empfehlen, sich mit diesen Möglichkeiten zu beschäftigen, wenn Sie Zeit haben. Allerdings stellt für mich Azure Container Registry (ACR) aus verschiedenen Gründen die beste Option dar, um meine Container-Images zu speichern:

- Erstens handelt es sich um ein privates Verzeichnis für Ihre Container-Images, sodass Sie sich keine Sorgen um einen möglichen unerwünschten Zugriff auf Ihre Anwendungsdateien und Konfigurationen machen müssen. Sie können dieselben RBAC-Mechanismen anwenden, die wir in Kapitel 6 besprochen haben. RBAC unterstützt Sie dabei, den Zugriff auf Ihre Images einzuschränken und zu kontrollieren.
- Zweitens bedeutet das Speichern Ihrer Container-Images in einem Verzeichnis in Azure, dass sich Ihre Images dort in den gleichen Rechenzentren wie die Infrastruktur befinden, die für den Betrieb Ihrer Container-Instanzen oder -Cluster verwendet wird (die wir in Abschnitt 19.4.1 betrachten). Obwohl Container-Images relativ klein sein sollten, oft nur Dutzende von MB groß, können sie sich aufsummieren, wenn Sie diese Images weiterhin aus einem Remote-Verzeichnis herunterladen.

ACR bietet auch integrierte Replikations- und Redundanzoptionen, mit denen Sie Ihre Container in der Nähe des Einsatzortes bereitstellen und für den Zugriff der Benutzer*in ausführen können. Dieser regionale Standort ist ähnlich wie die globale Replikation von Cosmos DB in Kapitel 10 aufgebaut. Dort werden diese Millisekunden gezählt und Ihren Kund*innen so die schnellstmögliche Zugriffszeit auf Ihre Anwendungen ermöglicht.

Wenn all dies spannend für Sie klingt, schauen Sie sich die ACR Quick Starts an, die mit Ihrem eigenen privaten Repository in wenigen Minuten einsatzbereit sind: http://mng.bz/04rj.

19.4 Azure Kubernetes Service

Der Betrieb einer einzelnen Container-Instanz ist eine großartige Option, sie bietet Ihnen allerdings nicht viel Redundanz oder Skalierbarkeit. Erinnern Sie sich, wie wir uns in den vorherigen Kapiteln des Buchs damit befasst haben, wie Sie mehrere Instanzen Ihrer Anwendung ausführen, Lastausgleiche durchführen und sie automatisch skalieren können? Wäre es nicht großartig, wenn das Gleiche mit Containern wäre? Genau an dieser Stelle benötigen Sie einen Container-Orchestrator.

Wie der Name schon sagt, verwaltet ein *Container-Orchestrator* Ihre Container-Instanzen, überwacht deren Zustand und kann bei Bedarf skalieren. Orchestratoren können und möchten oft über eine größere Zahl von Optionen verfügen. Auf dieser hohen Ebene liegt das Hauptaugenmerk auf der Handhabung aller beweglichen Bestandteile, die beim Betrieb einer hochverfügbaren, skalierbaren, containerbasierten Anwendung benötigt werden. Zu einigen wenigen Container-Orchestratoren zählen Docker Swarm und Distributed Cloud Operating System (DC/OS). Doch ein System hat hier eine dominante Stellung eingenommen und ist zum Go-to-Orchestrator der ersten Wahl geworden – Kubernetes.

Kubernetes begann als ein Google-geführtes und -gesponsertes Open-Source-Projekt, das aus unternehmensinternen Tools zur Containerorchestrierung entstanden ist. Kubernetes erfreut sich einer großen Akzeptanz in der Open-Source-Community und ist eines der größten und am schnellsten wachsenden Open-Source-Projekte auf GitHub. Viele große Technologieunternehmen, darunter Red Hat, IBM und Microsoft, beteiligen sich am Kernprojekt Kubernetes.

In diesem Abschnitt befassen wir uns mit der gleichen Beispiel-Webanwendung aus der vorherigen ACI-Übung, um eine redundante, skalierbare Bereitstellung in Kubernetes durchzuführen. Sie werden am Ende ein paar Komponenten haben, wie in Abbildung 19.8 dargestellt.

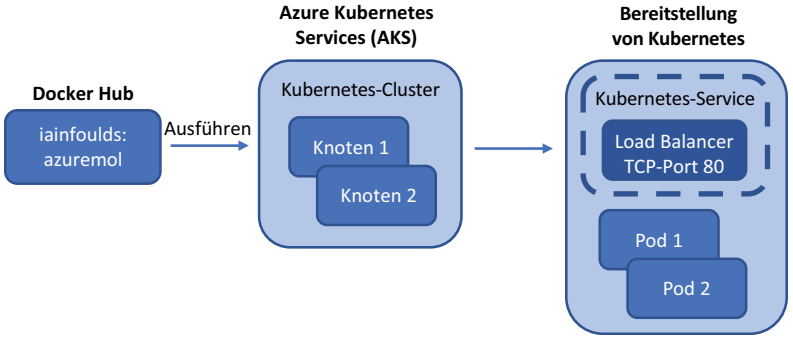


Abbildung 19.8 Ihr Beispiel-Container von Docker Hub läuft auf einem Kubernetes-Cluster mit zwei Knoten, den Sie im AKS erstellen. Die Kubernetes-Bereitstellung enthält zwei logische Pods, einen auf jedem Cluster-Knoten, wobei eine Container-Instanz in jedem Pod ausgeführt wird. Sie setzen dann einen öffentlichen Load-Balancer-Modul ein, damit Ihre Web App online angezeigt werden kann.

19.4.1 Erstellen eines Clusters mit Azure Kubernetes Services

In Kapitel 9 haben wir uns mit der Frage beschäftigt, wie die Größe von Virtual Machines die Komplexität der Bereitstellung und Konfiguration der zugrunde liegenden Infrastruktur reduziert. Sie legen fest, wie viele VM-Instanzen Sie in einem Scale-Set benötigen, und der Rest des Netzwerks, des Speichers und der Konfiguration wird für Sie bereitgestellt. AKS verfügt über eine ähnliche Funktionsweise, um einen robusten, skalierbaren Kubernetes-Cluster anzubieten, dessen Verwaltung über die Azure-Plattform erfolgt. Skalierungsgruppen können für die zugrunde liegenden VMs verwendet werden, die im AKS Cluster laufen, und diese VMs können über Verfügbarkeitszonen verteilt werden. Azure Load Balancer, auch zonenredundant, werden verwendet. Im Grunde genommen vereint AKS mehrere der Infrastrukturkomponenten und Best Practices, die Sie in diesem Buch bisher gelernt haben.

Jetzt testen

Führen Sie die folgenden Schritte aus, um die Informationen zu Ihrem AKS Cluster anzuzeigen:

1. Öffnen Sie das Azure Portal, und wählen Sie das Cloud Shell-Symbol aus dem oberen Menü.
2. Zu Beginn des Kapitels haben Sie einen Kubernetes Cluster erstellt. Der Prozess hat ein paar Minuten gedauert, aber hoffentlich ist er jetzt abgeschlossen! Betrachten Sie den Status des Clusters wie folgt:

```
az aks show \
  --resource-group azuremolchapter19 \
  --name azuremol
```

Der `provisioningState` gegen Ende sollte `Erfolgreich` melden.

3 Wenn Ihr Cluster bereit ist, erhalten Sie eine Datei mit Anmeldeinformationen, mit der Sie die Kommandozeilen-Tools in Kubernetes zur Authentifizierung und Verwaltung von Ressourcen verwenden können:

```
az aks get-credentials \
  --resource-group azuremolchapter19 \
  --name azuremol
```

Nichts weiter ist erforderlich, um Kubernetes in Azure zum Laufen zu bringen! Sie fragen sich vielleicht: „Kann ich nicht einfach meinen eigenen Cluster mit VMs oder Skalierungssets erstellen und die gleichen Docker- und Kubernetes-Komponenten manuell installieren?“. Natürlich können Sie das. Die Parallele hier ist der IaaS- und PaaS-Ansatz für VMs im Vergleich zu Web Apps. Der Web App-Ansatz bietet viele Vorteile: Sie müssen sich nur um die Konfigurationsmöglichkeiten auf hoher Ebene kümmern, und laden dann Ihren Anwendungscode hoch. Ein verwalteter Kubernetes-Cluster, wie er von AKS angeboten wird, reduziert den Grad der Komplexität und die Verwaltung. Ihr Schwerpunkt wird dadurch auf Ihre Anwendungen und die Erfahrung Ihrer Kunden verlegt.

Genauso wie Sie VMs anstelle von Web Apps bevorzugen können, können Sie auch Ihren eigenen Kubernetes-Cluster bereitstellen, anstatt AKS zu verwenden. Daran ist nichts auszusetzen, denn beide Ansätze verwenden am Ende die gleichen Azure Service-Komponenten. Mit VMs, Skalierungssätzen, Load Balancer und NSGs haben wir uns in den vorherigen Kapiteln befasst. Sie alle sind Bestandteile von AKS-Clustern, obwohl sie abstrahiert sind. Im Hinblick auf Planung und Fehlersuche sollten Sie die Funktionsweise verstehen können, um eine einwandfreie Funktion von verwalteten Kubernetes-Angeboten zu gewährleisten. Ihre Präferenzen und die Zeit, die Sie für die Verwaltung Ihrer Infrastruktur aufwenden möchten, bilden die Faktoren, mit denen Sie Ihren Entscheidungsprozess bei der Erstellung einer neuen Anwendung mit Containern in Azure steuern können.

19.4.2 Betrieb einer einfachen Website auf Kubernetes

Sie haben in Abschnitt 19.4.1 einen Kubernetes-Cluster erstellt, aber es wird noch keine Anwendung ausgeführt. Lassen Sie uns das ändern! Sie müssen die Kubernetes-Bereitstellung erstellen, die Sie zuvor in Abbildung 19.8 gesehen haben; siehe Abbildung 19.9.

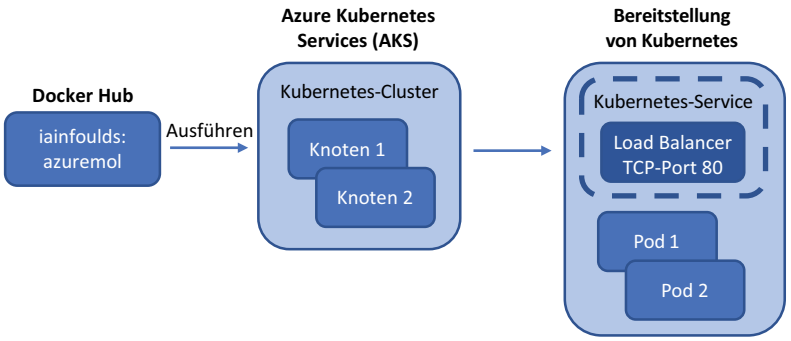


Abbildung 19.9 Mit dem in AKS erstellten Kubernetes-Cluster können Sie eine Kubernetes-Bereitstellung erstellen und Ihre App ausführen. Ihr Container wird dabei auf beiden Knoten mit einem logischen Pod auf jedem Knoten ausgeführt; Sie müssen einen Kubernetes-Dienst erstellen, der einen öffentlichen Load Balancer freigibt, um den Datenverkehr zu Ihrer App zu leiten.

Jetzt testen

Führen Sie die folgenden Schritte aus, um eine Anwendung in Ihrem Kubernetes-Cluster bereitzustellen:

1. Sie interagieren mit einem Kubernetes-Cluster über ein Befehlszeilenprogramm namens `kubectl`. Verwenden Sie das gleiche Container-Image iainfoulds/azuremol vom Docker Hub, das Sie als Container-Instanz ausgeführt haben:

   ```
   kubectl run azuremol \
     --image=docker.io/iainfoulds/azuremol:latest \
     --port=80
   ```

   Es kann etwa eine Minute dauern, bis das Container-Image vom Docker Hub heruntergeladen und die Anwendung in Kubernetes gestartet wird. Die Anwendung wird in einem *Pod* ausgeführt – ein logisches Konstrukt in Kubernetes, das jeden Container beherbergt.
2. Pods können zusätzliche Hilfskomponenten enthalten, aber im Moment können Sie den Status Ihres Containers überwachen, indem Sie sich den Pod ansehen:

   ```
   kubectl get pods --watch
   ```

Selbst wenn der Status des Pod als `Ausgeführt` gemeldet wird, können Sie nicht auf Ihre Anwendung zugreifen. Die Container-Instanz, die Sie zuvor erstellt haben, könnte den Datenverkehr über eine öffentliche IP-Adresse direkt an diese eine Instanz weiterleiten. Was ist aber Ihrer Meinung nach erforderlich, damit ein Kubernetes-Cluster den Datenverkehr an Container weiterleiten kann? Richtig – es ist „ein Load Balancer" – herzlichen Glückwunsch! Im Moment verfügen Sie nur über einen Pod – eine einzige Container-Instanz. Sie skalieren die Anzahl der Pods im in der praktischen Übung am Ende des Kapitels. Damit das funktioniert, benötigen Sie eine Möglichkeit, um den Datenverkehr an mehrere Instanzen weiterzuleiten. Also, sagen wir Kubernetes, er soll einen Load Balancer verwenden.

An dieser Stelle wird die Integration zwischen Kubernetes und Azure richtig aufregend. Wenn Sie Kubernetes auffordern, einen Load Balancer für Ihre Container zu erstellen, greift Kubernetes auf die Azure-Plattform zurück und erstellt einen Load Balancer in Azure. Dieser Azure Load Balancer ist derselbe, den wir uns in Kapitel 8 kennengelernt haben. Er ist mit Frontend- und Backend-IP-Pools und Load Balancer-Regeln ausgestattet, und Sie können Zustandsüberprüfungen konfigurieren. Wenn Ihre Kubernetes-Bereitstellung nach oben oder unten skaliert, wird der Load Balancer bei Bedarf automatisch aktualisiert.

Jetzt testen

Führen Sie die folgenden Schritte aus, um Ihre Anwendung im Internet verfügbar zu machen:

1. Teilen Sie Kubernetes mit, dass Sie einen Load Balancer verwenden möchten, und fügen Sie eine Regel zur Verteilung des Datenverkehrs auf Port 80 hinzu:

   ```
   kubectl expose deployment/azuremol \
     --type="LoadBalancer" \
     --port 80
   ```

2 Beobachten Sie wie bisher den Status Ihrer Service-Bereitstellung:

```
kubectl get service azuremol --watch
```

Mit der Zuweisung einer externen öffentlichen IP-Adresse beendet Azure Load Balancer die Bereitstellung. Damit ist die Herstellung einer Verbindung für Kubernetes-Cluster und Knoten abgeschlossen.

3 Öffnen Sie die öffentliche IP-Adresse Ihres Dienstes in einem Webbrowser, um die Ausführung Ihrer Webanwendung zu betrachten.

Anwendungsbereitstellungen in Kubernetes sind oft viel aufwändiger als in diesem einfachen Beispiel. Typischerweise definieren Sie ein Service-Manifest, ähnlich einer Resource Manager-Vorlage, das alle Merkmale Ihrer Anwendung bestimmt. Diese Eigenschaften können die Anzahl der auszuführenden Instanzen Ihrer Anwendung, jeden anzuschließenden Speicher, Lastausgleichsmethoden und Netzwerkports usw. beinhalten. In der Praxis tun Sie dies nicht einmal manuell. Ein CI/CD-System wie Azure DevOps oder Jenkins automatisiert die Bereitstellung von Anwendungen und Diensten direkt im AKS Cluster. Eine großartige Eigenschaft von AKS besteht darin, dass Sie sich keine Sorgen um die Installation und Konfiguration von Kubernetes machen müssen. Wie bei anderen PaaS-Diensten wie Web Apps und Cosmos DB bringen Sie Ihre Anwendungen mit und überlassen der Azure-Plattform die Handhabung der zugrunde liegenden Infrastruktur und Redundanz.

Aufrechterhaltung einer sauberen und ordentlichen Struktur

Denken Sie daran, Ihre Ressourcengruppen zu bereinigen und zu löschen, damit Sie nicht zu viele Ihrer kostenlosen Azure-Credits verbrauchen. In der Anfangsphase beim Kennenlernen von Containern müssen Sie unbedingt auch darauf achten, welche Azure-Ressourcen Sie eingeschaltet lassen. Eine einzelne Web App kostet nicht viel, aber ein fünfknotiger AKS-Cluster und ein paar Container-Instanzen mit georeplizierten Azure Container Registry-Images können zu einer wirklich kostenintensiven Angelegenheit werden!

ACI-Instanzen werden sekundengenau abgerechnet, und die Kosten summieren sich schnell, wenn sie für Tage oder Wochen in Betrieb bleiben. Ein AKS-Cluster führt eine VM für jeden Knoten aus. Wenn Sie also viele VMs in Ihrem Cluster hochskalieren und ausführen, bezahlen Sie bei einer VM für jeden Knoten.

Ein Gebühr für die Anzahl der Container, die jeder dieser AKS-Knoten ausführt, wird nicht erhoben. Allerdings wird wie bei jeder VM ein AKS-Knoten hohe Kosten verursachen, wenn er lange ausgeführt wird. Ein großer Vorteil von Kubernetes besteht darin, dass Sie Ihre Dienstkonfigurationen (die Definition für Ihre Pods, Load Balancer, Autoskalierung usw.) exportieren können, um sie an anderer Stelle einzusetzen. Beim Erstellen und Testen Ihrer Anwendungen müssen Sie keinen AKS-Cluster ausführen. Sie können einen Cluster nach Bedarf und Ihren Dienst aus einer früheren Konfiguration bereitstellen.

AKS-Cluster können hoch- und herunterskaliert werden, wie Sie in der Übung am Ende des Kapitels sehen werden. Sie können auch die automatische Skalierung konfigurieren, die diese Skalierung für Sie durchführt, je nach Last. Es ist die gleiche Art der automatischen Skalierung, die wir in Kapitel 9 für Skalierungsgruppen und Web Apps betrachtet haben. Sehen Sie schon, wie sich alles in Azure zusammenfügt?

Jetzt haben wir die Einführung in Container und Kubernetes abgeschlossen. Wir haben das Thema in Warp-Geschwindigkeit durchgenommen, Sie sollten sich deshalb keine Sorgen machen, wenn Sie sich gerade ein wenig überfordert fühlen! Manning hat mehrere großartige Bücher, wie *Learn Docker in a Month of Lunches*, von Elton Stoneman (https://livebook.manning.com/book/learn-docker-in-a-month-of-lunches)

(Fortsetzung)

und *Kubernetes in Action*, von Marko Luksa (https://livebook.manning.com/book/kubernetes-in-action), in denen Sie tiefere Einblicke zu Docker, Microservices-Anwendungsentwicklung und Kubernetes erhalten können. Wenn Sie die Inhalte dieses Kapitels spannend fanden und weitere Informationen erhalten möchten, dann bieten diese Bücher eine großartige Informationsquelle!

In den Beispielen in diesem Kapitel wurden Linux-VMs für die AKS-Cluster-Knoten verwendet und anschließend Linux-Container für NGINX ausgeführt. Container sind insofern etwas kompliziert, weil Sie Linux-Container beispielsweise nur auf Linux-Knoten ausführen können. Wie Sie zu Beginn des Kapitels gelernt haben, teilen sich Container das Gastbetriebssystem und den Kernel. Sie können also keine Windows-Container auf einem Linux-Knoten ausführen. Im Allgemeinen können Sie Linux-Container auch nicht auf einem Windows-Knoten ausführen. Es gibt einige technische Tricks, aber im Allgemeinen sollten der Container und das zugrunde liegende Knotenbetriebssystem übereinstimmen.

Das Tolle an AKS ist, dass Sie sowohl Linux- als auch Windows-Knoten betreiben können, also sowohl Linux- als auch Windows-Container! Sie müssen ein wenig darauf achten, wie diese verschiedenen Container auf den verschiedenen Node-Betriebssystemen geplant werden, aber dieser Ansatz erweitert die Palette der Anwendungen und Dienste, die Sie in AKS ausführen können, erheblich.

19.5 Übung: Skalieren Ihrer Kubernetes-Bereitstellungen

Beim einfachen Beispiel in diesem Kapitel wurde ein Kubernetes-Cluster mit zwei Knoten und einen einzelnen Pod erstellt, der Ihre Website betreibt. Bei dieser praktischen Übung erfahren Sie, wie Sie den Cluster und die Anzahl der Container-Instanzen skalieren können. Dies ist ein einfaches Beispiel, aber je mehr Knoten Sie haben, desto mehr Container-Instanzen können Sie betreiben, was besonders nützlich ist, je mehr Anwendungen Sie in Ihrem Cluster ausführen müssen.

1. Sie können mit `kubectl get nodes` anzeigen lassen, wie viele Knoten sich in Ihrem Kubernetes-Cluster befinden. Erhöhen Sie die Skalierung Ihres Clusters auf drei Knoten:

   ```
   az aks scale \
     --resource-group azuremolchapter19 \
     --name azuremol \
     --node-count 3
   ```

   Es dauert ein oder zwei Minuten, um die Skalierung zu erhöhen und den neuen Knoten hinzuzufügen.

2. Verwenden Sie `kubectl` erneut, um den Status Ihrer Knoten zu anzuzeigen. Wenn Sie einen Knoten skalieren, erstellt Kubernetes automatisch keine zusätzlichen Container-Instanzen für Ihre Anwendungen, sodass Sie nicht sofort einen Vorteil aus den zusätzlichen Compute-Ressourcen erhalten, die der neue Knoten bietet.

3 Betrachten Sie Ihre aktuelle Bereitstellung mit `kubectl get deployment azuremol`. Vorher wurde nur eine Instanz erstellt. Diese Beispielanwendung bildet nicht den größten Teil des neuen Knotens, den Sie dem Cluster in Schritt 1 hinzugefügt haben. Skalieren Sie auf bis zu fünf Instanzen oder *Replikate*:

```
kubectl scale deployment azuremol --replicas 5
```

4 Verwenden Sie `kubectl` erneut, um die Bereitstellung zu prüfen. Schauen Sie sich mit `kubectl get pods` die Pods und die ausgeführten Container-Instanzen an. All diese zusätzlichen Replikate wurden innerhalb weniger Sekunden gestartet und an den Load Balancer angeschlossen.

5 Verwenden Sie `kubectl get pods -o wide`, um zu sehen, auf welchen Knoten die Pods ausgeführt werden. Sehen Sie sich die letzte Ziffer im Knotennamen an, die angibt, welcher Knoten in der Skalierungsgruppe verwendet wird. Die Pods sollten auf alle Knoten in Ihrem Cluster verteilt werden. Da andere Anwendungen die Anzahl der Container auf ähnliche Weise skalieren würden, können Sie damit beginnen, die Verwendung der Compute-Ressourcen für alle Knoten im Cluster zu maximieren.

20 Azure und das Internet der Dinge

Einer der spannendsten Technologiebereiche der letzten Jahre ist für mich das Internet der Dinge (IoT). Ich glaube nicht ganz, dass eine Geschirrspülmaschine oder ein Kühlschrank noch mit dem Internet verbunden werden muss. Es bestehen außerdem berechtigte Bedenken hinsichtlich des Datenschutzes bei der Nutzung eines Fernsehers oder Audiogeräts, das ständig mit dem Internet verbunden ist und immer auf den Klang Ihrer Stimme hört, um einen Befehl zu empfangen. Es gibt jedoch viele weitere praktische Anwendungen für IoT-Geräte. Sie können die Produktionsausrüstung Berichte über ihren Zustand erstellen lassen, Wartungswarnungen generieren und es den Bedienern ermöglichen, ihre Effizienz in mehreren Fabriken auf der ganzen Welt zu verfolgen. Ein Speditionsunternehmen könnte die Telemetriedaten seiner Fahrzeuge über die beförderten Lasten und die durchschnittlichen Fahrzeiten übertragen und die Fahrer*innen bei Bedarf intelligent umleiten. Reedereien können mit dem Wissen zu Ihren Ressourcen jeden Container verfolgen und ihre Kund*innen bei einer effektiveren Verwaltung der Lieferkette unterstützen.

Bei Azure können Sie IoT-Geräte in eine Reihe von Diensten integrieren. Azure Web Apps können ein Frontend für die Visualisierung Ihrer Daten bereitstellen, Storage kann zum Protokollieren von Daten verwendet werden, die von Geräten übertragen werden, und Serverless-Funktionen wie Azure Logic Apps (die im nächsten und letzten Kapitel behandelt werden) können die empfangenen Daten verarbeiten.

In diesem Kapitel untersuchen wir, was IoT ist und wie Sie mit dem Azure IoT Hub Daten von Geräten zentral verwalten und erfassen können. Anschließend zeigen wir Ihnen, wie Sie eine Azure Web App verwenden können, um Daten von einem IoT-Gerät in Echtzeit anzuzeigen.

20.1 Was ist das „Internet der Dinge“?

Das Interesse am IoT hat in den letzten Jahren stark zugenommen, aber es ist ein vager Begriff, der sich auf viele Szenarien anwenden lässt. Auf grundlegender Ebene handelt es sich beim IoT um einen Ansatz, bei dem viele miteinander

verbundene Geräte – typischerweise kompakte, kostengünstige elektronische Geräte – mit zentralen Systemen und Anwendungen verbunden werden. Die vernetzten Geräte melden in der Regel Informationen, die sie von angeschlossenen Sensoren oder Dateneingängen sammeln. Diese Informationen können dann von einem zentralen System – möglicherweise mit KI oder ML, wie in Kapitel 17 beschrieben – verarbeitet, und es können dann entsprechende Maßnahmen durchgeführt werden. Abbildung 20.1 zeigt einen übergeordneten Ansatz für IoT.

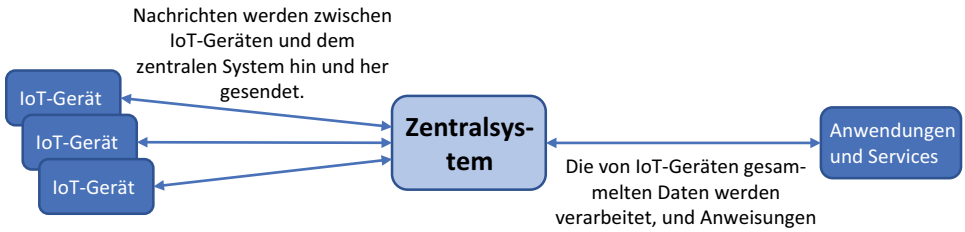


Abbildung 20.1 Nachrichten werden zwischen vielen vernetzten IoT-Geräten und einem Zentralsystem gesendet. Ihre Anwendungen und Dienste können dann die empfangenen Daten verarbeiten und Geräteanweisungen senden, um zusätzliche Aktionen als Reaktion auf die erfassten Daten einzuleiten.

Beispiele für IoT in der Praxis sind unter anderem die folgenden:

- *Parkhaus* – ein kleiner Sensor über jedem Parkplatz erkennt, ob ein Fahrzeug dort geparkt ist. Eine Leuchte über jedem Stellplatz leuchtet grün, wenn der Stellplatz frei, oder rot, wenn er belegt ist. Fahrer, die das Parkhaus betreten, sehen auf jeder Etage Echtzeit-Informationstafeln, die sie über die Zahl der offenen Parkplätze informieren. Die roten und grünen Leuchten über jedem Feld helfen dem oder der Fahrer*in, die Position der freien Plätze während der Fahrt entlang jedes Ganges schnell zu bestimmen.
- *Fabrik* – Maschinen in einer Fabrikhalle können Informationen über Betriebsleistung, Verbrauchsmaterial und Wartungsbedarf zurückmelden. Dann kann ein zentrales System eine*n Wartungstechniker*in einplanen, der die Geräte proaktiv repariert oder Verbrauchsmaterialien nachliefert, was die Ausfallzeiten in der Produktionslinie reduziert. In Kombination mit KI und ML können Wartungspläne vorab ermittelt werden, sodass die richtige Menge an Vorräten oder Rohstoffen geliefert wird, kurz bevor sie in der Produktion benötigt werden.
- *Transport* – Busse oder Bahnen des öffentlichen Nahverkehrs können mit GPS-Sensoren ausgestattet werden, die über Standort und Geschwindigkeit berichten. Es können auch Informationen zur Ticketausstellung gesammelt werden, um Berichte über die Zahl der transportierten Personen zu erstellen. Fahrgastinformationstafeln an einem Bahnhof oder Busbahnhof können dann in Echtzeit Auskunft über die Ankunftszeiten jedes Fahrzeugs erstatten. In Verbindung mit KI und ML können wartende Fahrgäste dann ausgehend von Verkehrsbedingungen, Verspätungen oder hohem Fahrgastaufkommen Vorschläge für alternative Strecken erhalten.

Das IoT arbeitet oft mit anderen Anwendungen und Diensten zusammen. Die Fabrik- und Transportszenarien könnten KI und ML nutzen, um Produktionsentscheidungen anhand eines besseren Informationsangebots zu treffen oder Fahrgästen Vorschläge zu unterbreiten. Webanwendungen können Informationen, die von IoT-Geräten empfangen werden, nutzen, um den Zugriff von mobilen Geräten aus zu ermöglichen

oder Warnmeldungen und Benachrichtigungen zu generieren. Die von IoT-Geräten empfangenen Daten können in einem Datenbanksystem wie Azure Cosmos DB protokolliert werden, die dann von Business Intelligence-Anwendungen verarbeitet werden, um Berichte zu generieren.

Weitere zukunftsweisende Ideen rund um das IoT umfassen Dinge wie Ihren Kühlschrank, der den Füllstand erfasst und eine Einkaufsliste erstellt oder sogar Lebensmittel in einem lokalen Lebensmittelgeschäft bestellt. Ihr Auto könnte Daten an den Händler zurückmelden, der alle benötigten Artikel oder Verbrauchsmaterialien bereit halten würde, wenn Sie das Fahrzeug zur Wartung bringen. Oder wie wäre es, wenn Ihr Wecker losgeht, um Sie morgens aufzuwecken, dann Ihre Kaffeemaschine einschaltet und alles für das Frühstück vorbereitet?

Ein großes Thema beim IoT ist die Gerätesicherheit. Bei vielen Geräten außerhalb Ihrer primären Netzwerkinfrastruktur, die oft mit dem öffentlichen Internet verbunden sind, besteht die Herausforderung in der Bereitstellung, Wartung und Aktualisierung dieser Geräte. Bei vielen IoT-Geräten handelt es sich um stromsparende, einfache elektronische Systeme. Sie verfügen möglicherweise nicht über die Speicher- oder Verarbeitungsfunktionen, um sich mit Sicherheits- und Anwendungsupdates zu aktualisieren, wie es bei herkömmlichen Desktop-PCs oder Laptops der Fall ist. Es reicht nicht aus, eine Reihe von IoT-Geräten, insbesondere Geräte für Verbraucher*innen, bereitzustellen, ohne einen Plan zur angemessenen Sicherung, Aktualisierung und Wartung zu haben.

Diese Sicherheitsbedenken sollten Sie nicht davon abhalten, Anwendungen und Dienste für IoT-Geräte zu entwickeln. IoT bringt eine Reihe neuer Herausforderungen für die traditionelle Gerätewartung mit sich. Allerdings stehen Ihnen Lösungen zur Verfügung, mit denen Sie Geräte zentral bereitzustellen und warten und die Gerätekommunikation sicher gestalten können.

Sie haben bestimmt schon vermutet, dass Azure eine solche IoT-Lösung anbietet! Und es bietet in der Tat eine Reihe von IoT-Diensten. Sehen wir uns an, wie Sie das Internet der Dinge (IoT) mit Azure erkunden können.

Beschleunigung Ihrer Azure IoT-Bereitstellungen

Dieses Kapitel befasst sich mit dem Azure IoT Hub – einem Dienst, mit dem Sie IoT-Geräte bereitstellen und anschließen können, um Ihre eigenen Lösungen zu entwickeln. Sie können festlegen, wie sich diese IoT-Geräte miteinander verbinden, welche Benutzer oder Anwendungen auf ihre Daten zugreifen können und eine sichere Verbindung gewährleisten. Wie Sie die Anwendungsinfrastruktur aufbauen und bereitstellen, um alles miteinander zu verbinden, liegt bei Ihnen.

Bei den Azure IoT Solution Accelerators handelt es sich um vorgefertigte Schlüsselszenarien, zu denen unter anderem die Fernüberwachung von Geräten oder einer angeschlossenen Fabrik zählen. Accelerators stellen gängige Azure-Dienste wie IoT Hub, Web Apps, Cosmos DB und Storage bereit und führen eine Beispielanwendung aus, die all diese verschiedenen Dienste integriert.

Sie müssen die Anwendung immer noch an Ihre eigene Umgebung, die verwendeten IoT-Geräte und die zu erfassenden und zu überwachenden Daten anpassen. Die Beschleuniger der IoT-Lösung bieten Ihnen hier ein hervorragendes Umfeld für den Einstieg. Mit dem IoT Hub können Sie IoT-Geräte mit Azure verbinden und es ermöglicht Ihnen anschließend die Bereitstellung zusätzlicher Dienste für Ihre Anforderungen. IoT-Lösungsbeschleuniger stellen vorgefertigte Lösungen bereit, die die gängigsten Azure-Dienste verwenden.

Wenn Sie nach der Lektüre dieses Kapitels zum IoT-Fan geworden sind und mehr darüber erfahren möchten, bieten Ihnen die Azure IoT Lösungsbeschleuniger eine gute Möglichkeit, um die Möglichkeiten in Azure besser kennenzulernen. Wie wir in diesem

Buch bereits erwähnt haben, handelt es sich bei Azure um weitaus mehr als nur ein oder zwei unabhängige Dienste. Sie können aus einer Vielzahl von Diensten wählen, die Sie gemeinsam bereitstellen können, um Ihren Kund*innen die bestmögliche Anwendungserfahrung zu bieten.

20.2 Zentrale Geräteverwaltung mit Azure IoT Hub

Mit dem Azure IoT Hub können Sie Daten von IoT-Geräten zentral verwalten, aktualisieren und streamen. Mit diesem Dienst können Sie Aktivitäten wie die Konfiguration von Anwendungsrouten für die von Geräten empfangene Daten, die Bereitstellung und Verwaltung von Zertifikaten zur Sicherung der Kommunikation und die Überwachung des Zustands mit Azure-Diagnosen und -Metriken durchführen. Sie können Ihre IoT-Geräte mit anderen Azure-Diensten und -Anwendungen verbinden, damit sie Daten als Teil einer umfassenderen Lösung senden und empfangen können. Wie bei allen Elementen in Azure kann der Zugriff mit RBAC gesteuert werden. Diagnosedaten können zentral gesammelt werden, um Fehler zu beheben und zu überwachen oder Warnungen zu erhalten. Abbildung 20.2 zeigt, wie ein IoT Hub die Funktion eines zentralen Standorts für IoT-Geräte übernimmt, um sich mit weiteren Azure-Diensten und -Anwendungen zu verbinden.

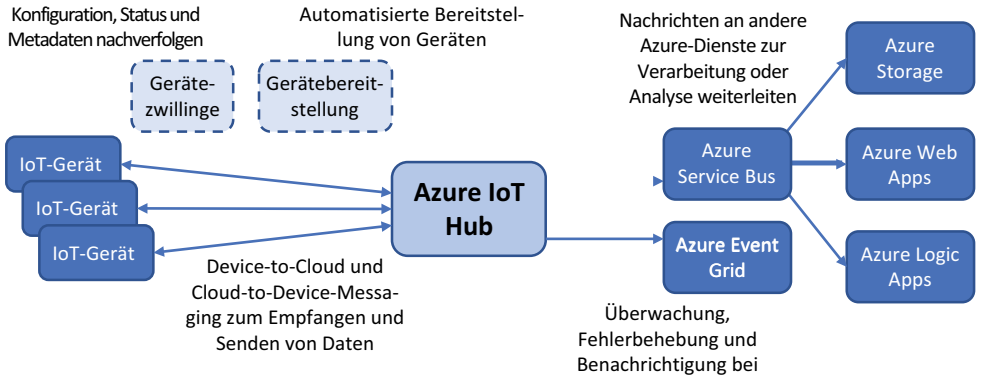


Abbildung 20.2 Mit einem IoT-Hub können Sie viele IoT-Geräte zentral und skalierbar bereitstellen und verwalten. Zwischen den Geräten und Azure besteht eine Zweiwegekommunikation zum Lesen und Aufzeichnen von Daten. Sie können von Geräten empfangene Daten verarbeiten und an andere Azure-Dienste wie Web Apps und Storage weiterleiten. Um Probleme zu überwachen und zu beheben, können Sie Informationen an das Azure Event Grid weiterleiten. Wir werden uns in Kapitel 21 mit dieser Anwendung befassen und dann auch auf andere Überwachungslösungen verweisen.

Sie steuern den Zugriff auf einen IoT-Hub mit Richtlinien für gemeinsamen Zugriff. Diese Richtlinien ähneln Benutzungskonten und Berechtigungen. Es gibt Standardrichtlinien, mit denen sich Geräte und Dienste mit dem IoT-Hub verbinden oder Informationen aus dem Geräteverzeichnis lesen und schreiben können, das die angeschlossenen IoT-Geräte und Sicherheitsschlüssel überwacht. Jeder Richtlinie kann eine oder mehrere der folgenden Berechtigungen zugewiesen werden:

- Verzeichnis lesen
- Verzeichnis schreiben

- Dienst verbinden
- Gerät verbinden

Gemeinsame Zugriffsschlüssel werden von Anwendungen und Diensten zur Verbindung mit einem IoT-Hub verwendet. Wie dies bei „Speicherung" in Kapitel 4 bereits angesprochen wurde, ermöglichen Ihnen die gemeinsamen Zugriffsschlüssel die Definition von Verbindungszeichenketten zur Identifizierung des Hosts, der Zugriffsrichtlinie und des Zugriffsschlüssels. Eine Verbindungszeichenfolge kombiniert den Zugriffsschlüssel, den Zugriffsrichtlinientyp und den Hostnamen des IoT-Hubs. Hier ist ein Beispiel für eine IoT-Hub-Verbindungszeichenfolge:

```
HostName=azuremol.azure-devices.net;SharedAccessKeyName=registryRead;
➥SharedAccessKey=6be2mXBVN9B+UkoPUMuwVDtR+7NZVBq+C7A1xCmQGAb=
```

Wir unterscheiden zwischen Primär- und Sekundärschlüsseln, die aus Sicherheitsgründen rotiert und aktualisiert werden können, ebenso wie die regelmäßige Aktualisierung von Kennwörtern. Lösungen wie Azure Key Vault, die in Kapitel 15 erläutert werden, bieten großartige Möglichkeiten, diese Schlüssel zu überwachen und zu speichern, damit sie Anwendungen bei Bedarf bereitgestellt werden können. Mit diesem Ansatz für Schlüsselverwaltung können Sie Zugriffsschlüssel häufig rotieren, ohne den gesamten Anwendungscode aktualisieren zu müssen.

Digitale Zertifikate können in einem IoT Hub gespeichert und auf IoT-Geräten automatisch bereitgestellt werden. Denken Sie daran, dass IoT-Geräte oft außerhalb Ihrer Kerninfrastruktur liegen und sich direkt über das Internet verbinden könnten, ohne jegliche sichere Netzwerkverbindung wie ein VPN. Stellen Sie sicher, dass alle Daten zwischen Ihren Geräten und dem IoT Hub mit SSL/TLS-Verbindungen verschlüsselt sind. Azure Key Vault kann SSL-Zertifikate generieren und speichern, die dann dem IoT Hub zugeordnet werden. Sie können auch eine bestehende Zertifizierungsstelle nutzen, um Zertifikate anzufordern und auszustellen. Wichtig ist, dass die gesamte Kommunikation zwischen Ihren IoT-Geräten und Azure verschlüsselt bleibt. Andernfalls erhalten Sie wahrscheinlich eine Fehlermeldung.

Mit den IoT Hub-Routen können Sie Daten von IoT-Geräten an andere Azure-Dienste senden. Sie können Kriterien wie z. B. Nachrichteninhalte definieren, die ein bestimmtes Schlüsselwort oder einen bestimmten Wert enthalten, und dann die Nachrichten weiterleiten. Sie werden anschließend in Azure Storage gespeichert oder von einer Webanwendung verarbeitet. In einer der folgenden Übungen simulieren Sie einen einfachen Temperatursensor, der an ein IoT-Gerät angeschlossen ist. Sie können im IoT-Hub eine Route definieren, um die eingehenden Daten zu überwachen. Wenn die aufgezeichnete Temperatur 30 °C überschreitet, sollen die Daten an eine Logic App weitergeleitet werden, um eine E-Mail-Benachrichtigung zu senden. In Kapitel 21 befassen wir uns mit der wunderbaren Welt der Serverless Computing- und Logic Apps!

Die Edge-Abläufe

In diesem Kapitel befassen wir uns mit dem Azure IoT Hub. Mit Azure IoT Edge – einem weiteren Dienst – können Sie einige Dienste wie Azure Functions und Stream Analytics in Ihrer lokalen Umgebung ausführen. Anstatt all Ihre IoT-Geräte Daten streamen zu lassen, die zentral in Azure verarbeitet werden sollen, können Sie die Daten an jedem beliebigen Standort verarbeiten.

Azure IoT Edge führt Anwendungen und Dienste in Containern aus (siehe Kapitel 19). Durch die Verwendung von Containern gewährleistet IoT Edge Portabilität und Konsistenz bei verschiedenen Geräten und Umgebungen. Vorgefertigte Azure-Dienste können bereitgestellt werden, oder Sie können Ihre eigenen Anwendungen schreiben und diese an Edge-Standorte verteilen.

Der Hauptvorteil von IoT Edge besteht darin, dass Sie einen Teil der Datenverarbeitung und der Netzwerkdatenübertragung entlasten. Wenn Sie Daten lokal in IoT Edge verarbeiten können, können Sie große Datenmengen stapeln und diese an Azure zurücksenden. Zentrale Anwendungen können dann Informationen von anderen Edge-Standorten sammeln, die von Diensten wie KI und ML verarbeitet werden sollen.

Ein weiterer großartiger Einsatzgebiet für Azure IoT Edge umfasst Remote-Standorte, die häufig in der Öl-, Gas- und Transportindustrie zu finden sind. Dort ist die Internetverbindung möglicherweise nicht ausreichend zuverlässig, um alle Daten der IoT-Geräte zur zentralen Verarbeitung an Azure zurückzusenden. Dank IoT Edge können diese Remote-Standorte weiterhin mit einem gewissen Maß an Autonomie funktionieren, selbst ohne eine Internetverbindung.

Wenn Sie eine Anwendungsinfrastruktur mit IoT-Geräten planen, sollten Sie prüfen, wie Sie mit Netzwerkausfällen und schlechten Internetverbindungen zurechtkommen. Bei einer Umgebung, die auf das Internet angewiesen ist, müssen Sie redundante Internetverbindungen und Geräte zur Weiterleitung der Daten einplanen. Alternativ können Sie IoT Edge nutzen, um Daten lokal zu verarbeiten, wenn dies nicht zentral in Azure möglich ist.

Jetzt testen

Um mit IoT zu beginnen und einen IoT Hub zu erstellen, führen Sie die folgenden Schritte aus:

1. Öffnen Sie das Azure Portal. Starten Sie Cloud Shell, und erstellen Sie eine Ressourcengruppe, z. B. `azuremolchapter20`:

   ```
   az group create --name azuremolchapter20 --location eastus
   ```

2. Sie haben in diesem Buch viel mit dem Azure CLI gearbeitet, denn mit den Befehlen Cloud Shell und CLI können Sie Ressourcen schnell erstellen und verwalten. Wie bereits in vorherigen Kapiteln erwähnt, kann Azure CLI auch zusätzliche Module verwenden, die als *Erweiterungen* bezeichnet werden. Diese Erweiterungen erhöhen die Funktionalität und aktualisieren oft außerhalb des regulären Release-Zyklus der zentralen Azure CLI. Azure IoT wächst schnell und erhält neue Funktionen, sodass die wichtigsten Befehle zur Interaktion mit dem IoT Hub von einer Azure CLI-Erweiterung stammen.

   Um die komplette Funktionalität zu nutzen, die Sie für diese Übungen benötigen, installieren Sie die Azure CLI IoT-Erweiterung:

   ```
   az extension add --name azure-cli-iot-ext
   ```

3. Wählen Sie Erstellen aus, und geben Sie einen Namen z. B. `azuremol` an. Bei diesen Übungen können Sie einen IoT-Hub aus dem kostenfreien Angebot f1 verwenden:

   ```
   az iot hub create \
     --resource-group azuremolchapter20 \
     --name azuremol \
     --sku f1 \
     --partition-count 2
   ```

HINWEIS Sie können nur einen Hub aus dem kostenlosen Angebot pro Abonnement erstellen. Diese Hubs eignen sich allerdings hervorragend zum Testen der Kommunikation zwischen Geräten und zur Integration mit anderen

Azure-Diensten. Der Hub aus dem kostenlosen Angebot ist derzeit auf 8.000 Nachrichten pro Tag beschränkt und unterstützt maximal 500 verbundene Geräte. Das mag nach viel klingen, aber abhängig von Ihrer Aktivität würde ein einzelnes Gerät, das etwa alle 12 Sekunden eine Nachricht an den IoT-Hub sendet, diese 8.000 Nachrichtengrenze überschreiten!

Ihr IoT Hub sieht im Moment ziemlich leer aus. In der Tat können Sie damit ohne ein oder mehrere angeschlossene IoT-Geräte nicht viel unternehmen. Ein gängiges Gerät, das für IoT-Anwendungen eingesetzt werden kann, ist Raspberry Pi. Dabei handelt es sich um einen kostengünstigen Minirechner, der an WLAN-Netzwerke angeschlossen werden kann und handelsübliche Sensoren für Temperatur, Feuchtigkeit und Druck verwendet. Sie können damit auch kleine Motoren, Leuchten und Timer steuern. Es ist aber nicht erforderlich, einen Raspberry Pi zu kaufen, um mit einem IoT-Hub zu arbeiten. Sie können ihn in Ihrem Webbrowser simulieren!

20.3 Erstellen eines simulierten Raspberry Pi-Geräts

IoT-Geräte bieten großartige Möglichkeiten, doch Sie müssen dazu das tatsächliche Gerät besitzen, richtig? Nicht wirklich! Es gibt verschiedene Möglichkeiten, ein IoT-Gerät mit Software zu simulieren. Mit diesem softwarebasierten Ansatz können Sie sich darauf konzentrieren, Ihre Anwendung schnell zu erstellen und diese dann auf echte Hardware zu übertragen. Sie müssen immer noch darauf achten, wie Ihr Code auf echter IoT-Hardware, insbesondere auf Geräten mit niedrigem Energieverbrauch, ausgeführt wird. Diese Geräte können möglicherweise nicht auf alle erforderlichen Bibliotheken oder sogar Speicherressourcen Ihrer simulierten Anwendung zugreifen.

Microsoft stellt einen kostenlosen Raspberry Pi-Simulator über GitHub zur Verfügung, auf den Sie unter https://azure-samples.github.io/raspberry-pi-web-simulator zugreifen können. Ein Raspberry Pi eignet sich hervorragend zum Testen, aber seien Sie vorsichtig, wenn Sie günstige Standardhardware wie den Raspberry Pi in Produktionsumgebungen einsetzen. Planen Sie, wie Sie solche Geräte aktualisieren und verwalten würden. Dedizierte Geräte wie Azure Sphere (https://azure.microsoft.com/services/azure-sphere) bieten zusätzliche Sicherheits- und Verwaltungsoptionen. Für dieses Buch und in Ihrer eigenen Test- und Lernumgebung ist der Raspberry Pi eine gute Alternative. In diesem Simulator, einem gängigen BME280-Sensor, der Temperatur- und Feuchtigkeitswerte erfasst, wird in der Software simuliert. Eine dazugehörige simulierte LED zeigt dabei an, wann das Gerät Daten an den IoT Hub sendet. Eine umfassende Anpassung ist hier nicht möglich, aber Sie können an dieser Stelle lernen, wie eine grundlegende Node.js-Anwendung auf dem Raspberry Pi ausgeführt werden, Daten von einem Sensor abfragen und diese an Azure zurücksenden kann.

HINWEIS Sie sollten sich keine Sorgen machen, wenn Sie mit Gebieten wie dem Raspberry Pi, Elektronik- und Temperatursensoren und Node.js nicht vertraut sind. Wie in den Kapiteln über KI und ML, Container und Kubernetes werden wir uns nicht umfassend mit IoT-Geräten und -Programmierung befassen. Wenn Sie das Gefühl haben, am Ende dieses Kapitels einen Lötkolben anschließen und zum Elektronik-Experten zu werden, sind Sie herzlich willkommen!

Vor der Verwendung eines Raspberry Pi-Simulators müssen Sie eine Gerätezuordnung im Azure IoT Hub erstellen. Dieser Prozess erzeugt eine eindeutige Geräte-ID, damit Ihr IoT-Hub weiß, mit welchem Gerät er kommuniziert und wie er die Daten verarbeitet. In komplexeren Szenarien könnten Sie zusätzliche Einstellungen für das Gerät vorgeben und digitale Zertifikate übertragen. Bei dieser Übung erstellen Sie einfach eine Geräteidentität.

Jetzt testen

Führen Sie die folgenden Schritte aus, um ein simuliertes Raspberry Pi-IoT-Gerät zu erstellen:

1. Erstellen Sie in der Azure Cloud Shell eine Geräteidentität in Ihrem IoT Hub, wie z. B. `azuremol`, und geben Sie einen Namen für das Gerät an, wie z. B. `raspberrypi`:

```
az iot hub device-identity create \
--hub-name azuremol \
--device-id raspberrypi
```

2. Erinnern Sie sich an die Richtlinien für gemeinsamen Zugriff aus Abschnitt 20.2? Jedes IoT-Gerät verfügt außerdem über einen eigenen Zugriffsschlüssel und eine Verbindungszeichenfolge, die es bei der Rückmeldung an Ihren IoT Hub identifizieren. Diese zentrale Funktion von Azure IoT schützt Geräte und minimiert das Risiko einer Gefährdung, wenn ein Gerät kompromittiert wird.

   Um Ihr Gerät mit dem Raspberry Pi-Simulator zu verwenden, benötigen Sie die Informationen der Geräteverbindungszeichenfolge. Diese eindeutige Kennung enthält den Host-Namen Ihres IoT-Hubs, die ID des Geräts und einen Zugriffsschlüssel:

```
az iot hub device-identity show-connection-string \
  --hub-name azuremol \
  --device-id raspberrypi \
  --output tsv
```

3. Kopieren Sie den Inhalt Ihrer Verbindungszeichenfolge, die Sie in Schritt 4 benötigen. Die Ausgabe kann wie folgt aussehen:

```
HostName=azuremol.azure-devices.net;DeviceId=raspberrypi;
➥SharedAccessKey=oXVvK40qYYI3M4u6ZLxoyR/PUKV7A7RF/JR9WcsRYSI=
```

4. Jetzt kommt der unterhaltsame Teil! Öffnen Sie den Raspberry Pi-Simulator in Ihrem Webbrowser: https://azure-samples.github.io/raspberry-pi-web-simulator. Betrachten Sie den Codebereich rechts im Simulator. Um Zeile 15 herum sollte die Variable `connectionString` zu sehen sein, die Sie zur Angabe *[Ihrer IoT-Hub-Geräteverbindungszeichenfolge]* auffordert. Kopieren und fügen Sie Ihre Verbindungszeichenfolge aus Schritt 3 ein, wie in Abbildung 20.3 dargestellt.
5. Wählen Sie die Schaltfläche „Run“ (Ausführen) direkt unter dem Codefenster, um den Simulator zu starten.

   Alle zwei Sekunden erscheint im Konsolenfenster eine Meldung, die die an den IoT-Hub gesendeten Daten anzeigt. Die rote LED im Schaltplan blinkt in diesem Fall ebenfalls und simuliert, wie die an den Raspberry Pi angeschlossenen Ausgänge gesteuert werden können. Die Ausgabemeldung im Konsolenfenster ähnelt der Folgenden:

```
Sending message: {"messageId":1,"deviceId":"Raspberry Pi Web
➥Client","temperature":24.207095037347923,
➥"humidity":69.12946775681091}
```

   Woher stammen die Temperatur- und Feuchtigkeitswerte? Bei diesem Gerät handelt es sich um einen simulierten Raspberry Pi. Es verfügt über keinen echten BME280-Sensor, sodass die Anwendung diese Werte in der Software generiert. Sehen Sie sich den Rest des Codes im Simulatorfenster an, um Zeile 99 herum

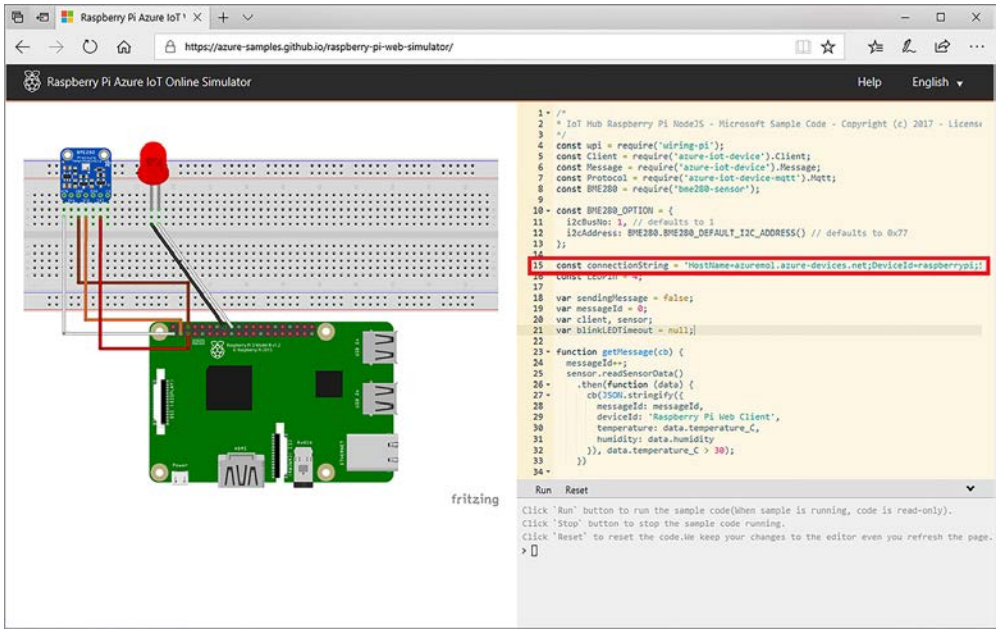


Abbildung 20.3 Kopieren und fügen Sie die Verbindungszeichenfolge für Ihr Azure IoT-Gerät in den Raspberry Pi-Simulator ein. Die Variable `connectionString` wird verwendet, um die simulierten Sensordaten an Azure zu übertragen.

definiert die Anwendung den Sensor. Der Simulator repliziert dann das Verhalten des realen Sensors und erzeugt Daten, die vom Sensor an die Anwendung zurückgegeben werden. Es ist ein einfaches Beispiel, und Sie könnten hier viele weitere Werte ablesen: Umdrehungen pro Minute (U/min) eines Motors, oder GPS-Koordinaten eines Versandbehälters oder LKWs, und so weiter. Hier besteht ein Gleichgewicht zwischen der Simulation eines Geräts innerhalb der Software und der Erstellung einer funktionalen Anwendung mit realen Hardware- und Sensordaten. Wenn Sie sich mit Azure IoT näher befassen möchten, müssen Sie die Geräte irgendwann erwerben oder leihen.

6 Überprüfen Sie den Quotenstatus, um zu bestätigen, dass Ihre simulierten Gerätenachrichten vom IoT Hub empfangen werden. Geben Sie den Namen Ihres IoT Hubs ein, z. B. azuremol:

```
az iot hub show-quota-metrics --name azuremol
```

Die Ausgabe ähnelt dem folgenden Beispiel, bei dem 5 von den maximal 8.000 Gesamtnachrichten pro Tag empfangen wurden und 1 angeschlossenes Gerät von maximal 500 Gesamtgeräten vorhanden ist. Es kann einige Minuten in Anspruch nehmen, bis diese Metriken mit Daten gefüllt sind. Sie sollten sich deshalb keine Sorgen, wenn Sie keine Daten sehen:

```
[
  {
    "currentValue": 5,
```

```
    "maxValue": 8000,
    "name": "TotalMessages"
  },
  {
    "currentValue": 1,
    "maxValue": 500,
    "name": "TotalDeviceCount"
  }
]
```

Sie können auch im Azure Portal suchen: Wählen Sie Ihre Ressourcengruppe und dann Ihren IoT Hub. Auf der Seite „Overview“ (Übersicht) meldet die Hub-Nutzung die Anzahl der empfangenen Nachrichten und angeschlossenen Geräte an. Auch hier kann es ein oder zwei Minuten dauern, bis die Nachrichten erscheinen und im Verhältnis zum Kontingent aufgezeichnet werden. Jede Anwendung könnte die empfangenen Nachrichten sofort nutzen, wie wir im Abschnitt 20.4 sehen werden.

Ärger im Paradies

Wenn Sie keine Nachrichten in Ihrem IoT Hub empfangen, überprüfen Sie das Ausgabefenster Ihres simulierten Raspberry Pi-Geräts. Eine der ersten Aktivitäten der Anwendung ist die Herstellung der Verbindung zum Azure IoT Hub. Ein Verbindungsfehler wird angezeigt, wenn Ihre Verbindungszeichenfolge falsch ist. Achten Sie darauf, dass Sie die gesamte Verbindungszeichenfolge korrekt kopieren und einfügen. Die Verbindungszeichenfolge beginnt mit `HostName`, und das letzte Zeichen in jedem Zugriffsschlüssel ist immer ein Gleichheitszeichen (=).

Wenn das Ausgabefenster einen Fehler meldet, kopieren Sie den Fehlertext in Ihre bevorzugte Suchmaschine und suchen Sie nach einem passenden Ergebnis. Stellen Sie sicher, dass Sie keine der anderen Codezeilen geändert haben. Dies würde dann nämlich ein wirkliches Problem darstellen! Das Einzige, was Sie im Code-Fenster ändern müssen, ist die Zeile für Ihre Verbindungszeichenfolge.

Da das simulierte Raspberry Pi-Gerät in einem Webbrowser ausgeführt wird, kann es zu einem allgemeinen Website-Problem kommen. Versuchen Sie, die Seite zu aktualisieren, oder greifen Sie auf den Simulator in einem anderen Browser (https://azure-samples.github.io/raspberry-pi-web-simulator) zu.

20.4 Streamen von Azure IoT Hub-Daten zu Azure Web Apps

Ein Gerät, das sich mit einem IoT Hub verbindet, ist nicht zu gebrauchen, wenn es mit den Daten nichts anfangen kann. Hier können Sie beginnen, viele der Dienste und Funktionen zu integrieren, die Sie in diesem Buch kennengelernt haben. Möchten Sie an Azure Storage-Tabellen oder Warteschlangen streamen? Das schaffen Sie. Daten von IoT-Geräten in Azure VMs oder Containern verarbeiten? Fahren Sie direkt fort! Verwenden Sie Azure Cosmos DB, um Ihre Daten zu replizieren und dann mit global redundanten Azure-Web Apps und Traffic Manager darauf zuzugreifen? Natürlich!

Im Beispielszenario bildet der IoT Hub den Verbindungsmechanismus und Einstiegspunkt für Ihre IoT-Geräte in Azure. Der Hub selbst unternimmt nichts direkt mit den Daten. Für Ereignisse existiert ein Standard-Endpunkt, der einen großen Behälter für alle vom IoT-Gerät empfangenen Nachrichten bildet. Ihr simuliertes Raspberry Pi-Gerät sendet Nachrichten an den IoT Hub, und diese Meldungen haben den Ereignis-Endpunkt erreicht. Der Nachrichtenfluss von Geräten über den IoT Hub zu einem Endpunkt ist in Bild 20.4 dargestellt.

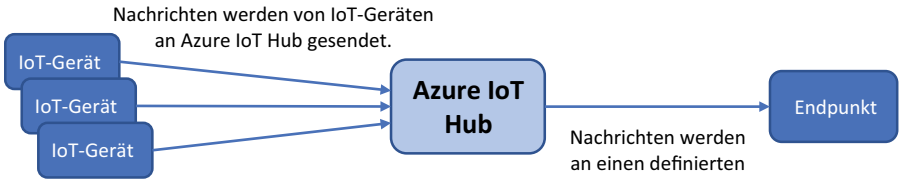


Abbildung 20.4 Ein IoT Hub empfängt Nachrichten von angeschlossenen IoT-Geräten und sendet die Nachrichten an einen Endpunkt. Diese Endpunkte können von anderen Azure-Diensten verwendet werden, um Daten von den IoT-Geräten zu nutzen. Es ist ein Standard-Endpunkt für Ereignisse vorhanden, von dem aus Dienste wie Web Apps Daten ablesen können.

Sie können benutzerdefinierte Endpunkte erstellen, die Nachrichten direkt an Azure-Dienste wie Storage und Service Bus weiterleiten. In Kapitel 4 haben wir uns mit Azure Storage-Warteschlangen beschäftigt, um eine Möglichkeit zu finden, Nachrichten zwischen Anwendungen hin und her zu leiten. Azure Service Bus ist eine robustere und skalierbarere Enterprise Messaging-Plattform. Dem Service Bus können Nachrichten hinzugefügt werden, wie z. B. Daten von IoT-Geräten. Andere Anwendungen können dann auf diese Nachrichten warten und entsprechend reagieren.

Wenn auf das Lesen von Nachrichten aus einer Nachrichtenschlange wie einem Service Bus aufgrund der Komplexität verzichten möchten, können Sie Verbrauchergruppen mit dem Standard-Ereignisendpunkt nutzen. Mit einer Verbrauchergruppe können Dienste wie Azure Web Apps, Daten vom Endpunkt abzulesen, wie in Abbildung 20.5 dargestellt. Jeder Dienst, der von Azure IoT Hub liest, sollte über eine eigene Verbrauchergruppe verfügen. Mehrere Dienste, jeder mit einer eigenen Verbrauchergruppe, können die gleichen Nachrichten empfangen und bei Bedarf verarbeiten.

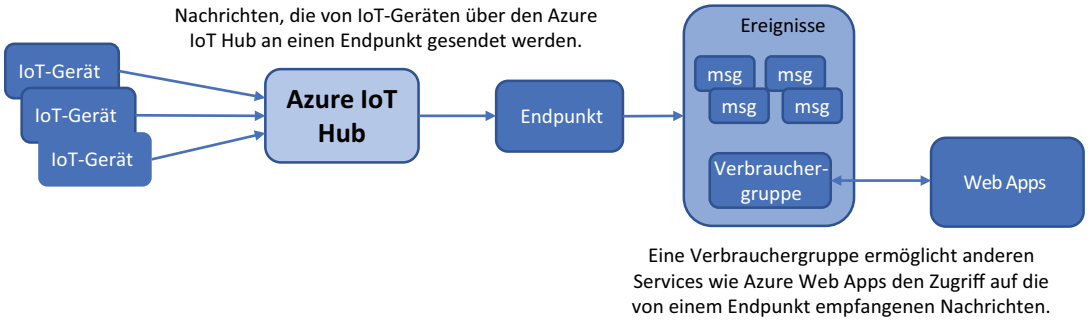


Abbildung 20.5 Nachrichten werden von IoT-Geräten an den IoT Hub gesendet, der die Nachrichten dann an einen Endpunkt weiterleitet. Verbrauchergruppen können in jedem Endpunkt angelegt werden. Diese Verbrauchergruppen ermöglichen anderen Azure-Diensten den Zugriff auf die Gerätemeldungen, auf die sie sonst keinen Zugriff hätten. Bei Verbrauchergruppen müssen Sie keine Nachrichtenwarteschlangen verwenden, damit externe Anwendungen IoT-Gerätedaten ablesen können.

Erstellen wir eine Azure Web App, die eine Verbrauchergruppe verwendet, um Nachrichtendaten in Echtzeit von Ihrem simulierten Raspberry Pi-Gerät abzulesen. Dieses einfache Beispiel zeigt, wie Sie Daten von IoT-Geräten streamen und über Webanwendungen darauf zugreifen können.

Jetzt testen

Führen Sie die folgenden Schritte aus, um eine Azure Web App zu erstellen, die Daten von IoT-Geräten abliest:

1. Erstellen Sie einen Azure App Service Plan für Ihre Web App in der Azure Cloud Shell, und geben Sie einen Namen an, z. B. `azuremol`. Bei diesen Übungen können Sie die kostenlose Stufe (f1) nutzen und so die Kosten niedrig halten:

```
az appservice plan create \
  --resource-group azuremolchapter20 \
  --name azuremol \
  --sku f1
```

2. Erstellen Sie Ihre Web App. Geben Sie einen Namen an, z. B. molwebapp, und aktivieren Sie ihn für die Verwendung mit Git, damit Sie die Beispielanwendung bereitstellen können. Wie bei anderen öffentlich zugänglichen Azure-Ressourcen müssen Sie auch einen eigenen, global eindeutigen Namen angeben.

```
az webapp create \
  --resource-group azuremolchapter20 \
  --plan azuremol \
  --name molwebapp \
  --deployment-local-git
```

3. Sie müssen die Verbrauchergruppe für Ihren IoT Hub sowie einige Einstellungen für Webanwendungen definieren. Mit diesen Einstellungen kann sich Ihre Web App mit Ihrem IoT Hub verbinden. Abbildung 20.6 zeigt, wie Sie in den nächsten Schritten fortfahren.

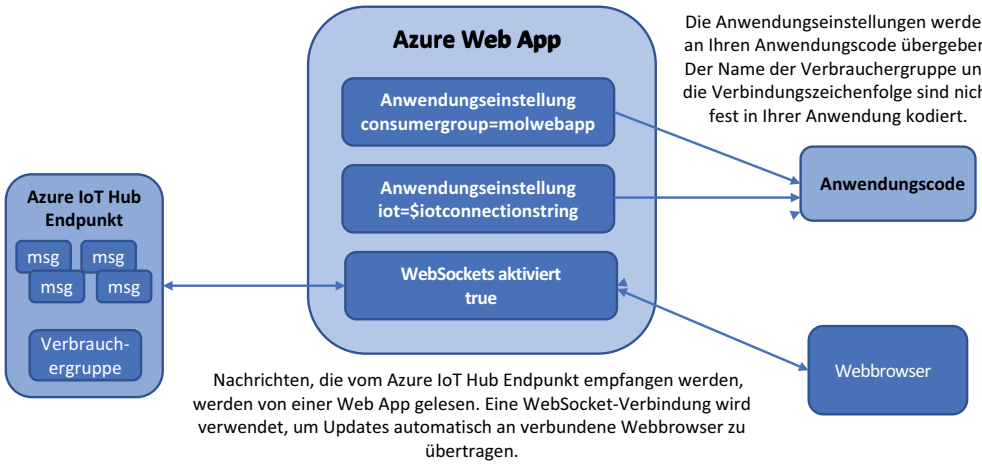


Abbildung 20.6 Damit Ihre Web App die Daten Ihres simulierten Raspberry Pi IoT-Geräts ablesen kann, legen Sie im IoT Hub eine Verbrauchergruppe an. Anschließend definieren Sie zwei Anwendungseinstellungen für Ihre Web App, mit denen Sie eine Verbindung zur Verbrauchergruppe herstellen können. Damit Ihr Webbrowser den Datenstrom vom Raspberry Pi beim Empfang neuer Daten automatisch empfangen kann, aktivieren Sie auch eine Einstellung für WebSockets.

4 Erstellen Sie eine Verbrauchergruppe, mit der Ihre Webanwendung auf die von Ihrem IoT-Gerät gestreamten Ereignisdaten zugreifen kann. Geben Sie Ihren IoT Hub an, wie z. B. `azuremol`, und geben Sie dann einen Namen für Ihre Verbrauchergruppe an, wie z. B. `molwebapp`. Stellen Sie sicher, dass Sie in den nächsten Schritten Ihren eigenen Namen verwenden. Ihre Verbrauchergruppe wird im Standard-Ereignisendpunkt erstellt:

```
az iot hub consumer-group create \
  --hub-name azuremol \
  --name molwebapp
```

5 Sie müssen Ihrer Web App den Namen der Verbrauchergruppe mitteilen. Erstellen Sie eine Einstellung für eine Webanwendung, die von der von Ihnen am Ende der Übung bereitgestellten Beispielanwendung verwendet wird. Mit Anwendungseinstellungen in Web Apps können Sie spezifische Einstellungen wie den Namen der Verbrauchergruppe und die Verbindungszeichenfolge definieren, ohne dass diese Werte in Ihrer Anwendung fest programmiert werden.

Geben Sie den Namen der Verbrauchergruppe am, die in Schritt 4 erstellt wurden, wie z. B. `mol-webapp`:

```
az webapp config appsettings set \
  --resource-group azuremolchapter20 \
  --name molwebapp \
  --settings consumergroup=molwebapp
```

6 Um sich mit Ihrem IoT Hub zu verbinden, muss Ihre Webanwendung die Verbindungszeichenfolge für den Hub kennen. Diese Verbindungszeichenfolge unterscheidet sich von derjenigen, die Sie in der vorherigen Übung für Ihr simuliertes Raspberry Pi-Gerät kopiert haben. Denken Sie daran, dass es eine Verbindungszeichenfolge für Ihren IoT Hub gibt, die gemeinsame Zugriffsrichtlinien verwendet, um Zugriffsberechtigungen zu definieren. Berücksichtigen Sie auch, dass es eine Verbindungszeichenfolge für jedes IoT-Gerät gibt. Ihre Web App muss Daten aus der IoT Hub-Endpunkt-Verbrauchergruppe ablesen. Sie müssen daher eine Verbindungszeichenfolge für den IoT Hub selbst definieren.

7 Holen Sie sich die IoT Hub-Verbindungszeichenfolge und weisen Sie diese einer Variablen namens `iotconnectionstring` zu, die in Schritt 8 verwendet wird:

```
iotconnectionstring=$(az iot hub show-connection-string \
  --hub-name azuremol \
  --output tsv)
```

8 Erstellen Sie eine weitere Anwendungseinstellung für die Web App, diesmal für den Verbindungsstring des IoT Hubs. Die in Schritt 7 definierte Variable wird verwendet, um die Beispielanwendung mit dem IoT-Gerät zu verbinden und Daten aus ihm abzulesen:

```
az webapp config appsettings set \
  --resource-group azuremolchapter20 \
  --name molwebapp \
  --settings iot=$iotconnectionstring
```

9 Aktivieren Sie WebSockets. Bei einem *WebSocket* handelt es sich um ein Zweiwege-Kommunikationsmittel zwischen Browser und Server. Die Beispielanwendung aktualisiert den Webbrowser automatisch mit den vom Raspberry Pi-Gerät empfangenen Daten. Zur Durchführung dieses automatisierten Updates

verwendet die Anwendung WebSockets. Der Server kann dann Daten an den Browser übertragen und ihn automatisch aktualisieren:

```
az webapp config set \
--resource-group azuremolchapter20 \
--name molwebapp \
--web-sockets-enabled
```

Lassen Sie uns hier eine Pause einlegen und darüber sprechen, womit wir uns bisher befasst haben. In vielen der vorherigen Kapitel haben Sie mit Web Apps gearbeitet, aber die Einstellungen für Webanwendungen und WebSockets sind neu für Sie. Abbildung 20.7 zeigt, wie Ihre Web App und Ihr IoT Hub verbunden sind.

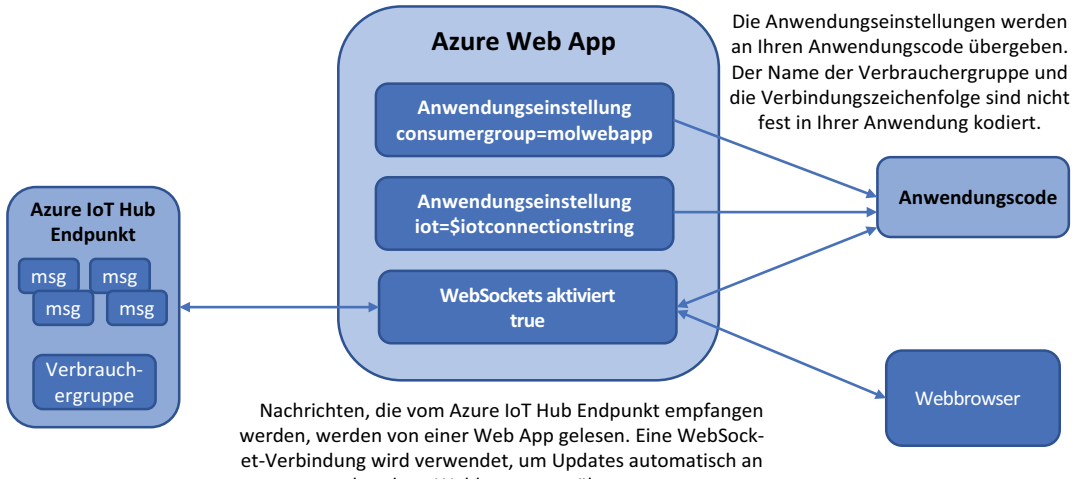


Abbildung 20.7 Wenn Nachrichten von IoT-Geräten gesendet werden, durchlaufen sie den IoT Hub auf ihrem Weg zu einem Endpunkt. Ihr Anwendungscode liest die Einstellungen der Webanwendung ein, die die IoT Hub-Verbindungszeichenfolge und die zu verwendende Verbrauchergruppe definieren. Nach der Verbindung der Anwendung mit dem IoT Hub ermöglicht die Consumer-Gruppe Web Apps das Ablesen der IoT-Gerätenachrichten. Jedes Mal, wenn eine neue Nachricht von einem IoT-Gerät empfangen wird, verwendet Ihre Web App eine WebSocket-Verbindung mit Webbrowsern, die auf Ihre Website zugreifen, um automatisch Updates zu übertragen. Mit dieser Verbindung können Sie die von IoT-Geräten übertragenen Echtzeitdaten anzeigen, wie z. B. Temperatur- und Feuchtigkeitsinformationen von Ihrem simulierten Raspberry Pi-Gerät.

Nun beenden wir die Übung und stellen die Beispielanwendung aus dem GitHub-Repository in Ihrer Web App bereit. Sie können dann die Web App in Ihrem Browser öffnen und die Echtzeitdaten sehen, die von Ihrem simulierten Raspberry Pi gestreamt werden!

10 Klonen Sie gegebenenfalls die GitHub-Repository-Beispiele wie folgt in Ihrer Cloud Shell:

```
git clone https://github.com/fouldsy/azure-mol-samples-2nd-ed.git
```

11 Wechseln Sie in das Verzeichnis für Kapitel 20:

```
cd azure-mol-samples-2nd-ed/20
```

12 Initialisieren Sie das Git Repository, und fügen Sie eine einfache Webseite hinzu:

```
git init && git add . && git commit -m "Pizza"
```

13 Um die Beispielanwendung hochzuladen, erstellen Sie eine Verbindung zu Ihrer Web App. Der folgende Befehl ruft das Web-App-Repository ab und konfiguriert das Git-Repository Ihrer lokalen Samples zur Herstellung einer Verbindung:

```
git remote add molwebapp \
  $(az webapp deployment source config-local-git \
  --resource-group azuremolchapter20 \
  --name molwebapp \
  --output tsv)
```

In den vorherigen Kapiteln habe ich Sie Ihnen die Aufgabe gegeben, nach dieser Adresse zu suchen. Inzwischen hoffe ich, dass Sie sich bereits aktiv mit weiteren von Azure CLI beschäftigt haben und festgestellt haben, dass viele dieser Informationen leicht zugänglich sind:

14 Verschieben Sie die HTML-Beispielseite mit dem folgenden Befehl zu Ihrer Webanwendung:

```
git push molwebapp master
```

15 Geben Sie bei Aufforderung das von Ihnen erstellte Kennwort für den Git-Benutzenden ein, den Sie in früheren Kapiteln verwendet haben (das in Kapitel 3 erstellte Konto).

Für den Fall, dass Sie Ihr Git-Kennwort nicht auf einer Haftnotiz notiert haben

Wenn Sie das Kennwort vergessen haben, können Sie es zurücksetzen. Als Erstes erhalten Sie den Benutzungsnamen Ihres lokalen Git-Bereitstellungskontos:

```
az webapp deployment user show --query publishingUserName
```

Um das Kennwort zurückzusetzen, geben Sie den Namen Ihres Kontos aus dem vorherigen Befehl ein. Folgen Sie dann den Aufforderungen, um ein neues Kennwort festzulegen. Das folgende Beispiel setzt das Kennwort für das Benutzungskonto namens azuremol zurück:

```
az webapp deployment user set --user-name azuremol
```

16 Zeigen Sie den Host-Namen für Ihre Web App an, und öffnen Sie die Adresse dann in einem Webbrowser:

```
az webapp show \
  --resource-group azuremolchapter20 \
  --name molwebapp \
  --query defaultHostName \
  --output tsv
```

Beim ersten Öffnen der Website in Ihrem Webbrowser kann es einige Sekunden dauern, da sich die Web App mit dem IoT Hub verbindet, die WebSocket-Verbindung startet und auf den Empfang der ersten Gerätenachricht wartet. Der Webbrowser sollte alle zwei Sekunden automatisch mit den neuesten simulierten

Daten vom Raspberry Pi-Gerät aktualisiert werden, wie in Abbildung 20.8 dargestellt.

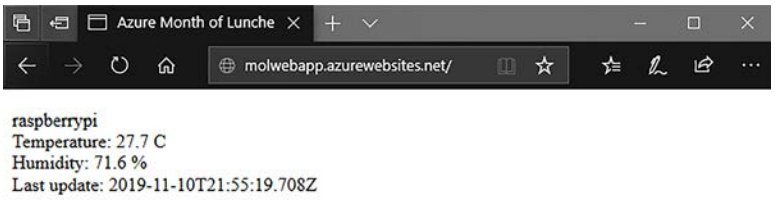


Abbildung 20.8 Die Beispielanwendung verwendet eine WebSocket-Verbindung zwischen Ihrem Webbrowser und der Webanwendung, um alle zwei Sekunden automatisch die neuesten Daten von Ihrem simulierten Raspberry Pi-Gerät zu aktualisieren.

Wenn Ihre Web App-Instanz keine Daten anzeigt, prüfen Sie, dass das simulierte Gerät Raspberry Pi noch eingeschaltet ist. Starten Sie bei Bedarf das simulierte Gerät und stellen Sie sicher, dass es mit dem Azure IoT verbunden ist und Nachrichten sendet. Die Daten sollten dann in Ihrer Web App-Instanz erscheinen.

20.5 Azure IoT-Komponentenprüfung

Ich hoffe, dass die Übungen in diesem Kapitel Ihnen einen Eindruck davon vermittelt haben, welche Dienste bei Azure für IoT-Lösungen verfügbar sind:

- *Azure IoT Hub* bietet eine hervorragende Möglichkeit, eine Vielzahl von IoT-Geräten bereitzustellen, zu verbinden und zu verwalten und diese dann in andere Azure-Dienste zu integrieren.
- *Azure IoT Solution Accelerators* bieten vorkonfigurierte Szenarien, die automatisch viele verschiedene Azure-Dienste integrieren, um eine komplette Anwendungsumgebung bereitzustellen.
- Mit *Azure IoT Edge* können Sie Azure-Dienste in Ihrer lokalen Umgebung bereitstellen, um Daten von IoT-Geräten zu verarbeiten, ohne alle Daten zentral an Azure zurückstreamen zu müssen.

Um ein besseres Verständnis für die Funktion von Azure IoT und IoT-Geräten im Allgemeinen zu erhalten, empfehle ich Ihnen, ein einfaches Raspberry Pi oder ein ähnliches Gerät zu erwerben. Sie sind relativ preiswert und werden oft mit ein paar verschiedenen Basissensoren oder elektrischen Komponenten geliefert, um verschiedene Ideen zu testen. Sie bieten Ihnen auch eine großartige Lernplattform, mit der Sie erkennen können, was bei der Integration von Hard- und Software möglich ist. Denken Sie aber an meine Warnungen aus Kapitel 17 zum Thema KI und ML und den Bau von Skynet! Manning bietet auch einige ausgezeichnete Bücher, wie z. B. *Building the Web of Things*, von Dominique D. Guinard und Vlad M. Trifa (https://www.manning.com/books/building-the-web-of-things) und *JavaScript on Things* von Lyza Danger Gardner (https://www.manning.com/books/javascript-on-things), die näher auf den Raspberry Pi, IoT Best Practices, JavaScript und Node.js Programmierung bei IoT-Geräten eingehen.

Erinnern Sie sich an meinen Hinweis, Ihre Ressourcengruppen immer zu löschen?

Die beste Vorgehensweise bei diesem Buch besteht darin, Ihre Ressourcengruppen am Ende jedes Kapitels zu löschen. Dieser Ansatz stellt sicher, dass Sie keine Dienste und Anwendungen ausführen, die Kosten verursachen, wenn Sie diese nicht nutzen.

Azure IoT bietet Ihnen eine großartige Plattform, um Daten zu Azure zu übertragen. In der Regel müssen Sie diese Daten verarbeiten und nicht nur in einer Web App anzeigen, wie Sie es in den Übungen getan haben. Kapitel 21 befasst sich mit Serverless Computing bei Logic Apps- und Functions-Diensten.

Die effektive Zusammenarbeit dieser Azure-Dienste wird in diesem Kapitel demonstriert. Löschen Sie deshalb die Ressourcengruppe und die Dienste nicht, die Sie in diesem Kapitel bereitgestellt haben. Diese verwenden Sie gleich zu Beginn von Kapitel 21. Auf der Grundlage der von Ihren IoT-Geräten empfangenen Daten können Sie dann erkennen, welche Maßnahmen Sie ergreifen können. Kehren Sie zu Ihrem simulierten Raspberry Pi-Gerät zurück, und wählen Sie die Schaltfläche „Stop“ (Unterbrechen). Andernfalls wird das Limit mit 8.000 Meldungen ziemlich schnell aufgebraucht sein!

20.6 Übung: Ausprobieren von IoT-Beispielanwendungen

In diesem Kapitel wurden viele neue Aspekte angesprochen. Ohne ein echtes IoT-Gerät verfügen Sie jedoch über eingeschränkte Möglichkeiten. Kapitel 21 nutzt Azure IoT Hub und die simulierten Raspberry Pi-Anwendung als Grundlage, sodass ich jetzt keine weiteren umfassenden Konfigurationen verwenden möchte. Hier sind ein paar weitere Aspekte, mit denen Sie sich beim Thema IoT befassten könnten:

1. In welchen Bereichen könnten Ihrer Meinung nach IoT-Geräte für Ihr Unternehmen von Vorteil sein? Wenn Sie aktuell nicht in einem Unternehmen beschäftigt sind, dann überlegen Sie sich ein Beispiel für das fiktive Pizzageschäft aus *Azure in der Mittagspause*.
2. Was können Sie tun, um die Situation für Ihre Kund*innen mit IoT zu verbessern?
3. Würden Sie Azure IoT Edge verwenden? Warum oder warum nicht?
4. Welche weiteren Azure-Dienste würden Sie integrieren, um Ihre Anwendungen auszuführen?
5. Wenn Sie in Ihrer Mittagspause noch Zeit haben, testen Sie einen der Azure IoT Solutions Accelerators von www.azureiotsolutions.com/Accelerators. Mit einem Device Simulation-Szenario können Sie eine VM und simulierte Sensoren wie beim simulierten Raspberry Pi-Gerät erstellen, allerdings mit einer viel umfassenderen Funktionspalette! Es dauert einige Minuten, um alle erforderlichen Ressourcen bereitzustellen. Schauen Sie sich dann im Azure Portal unbedingt an, was erstellt wurde und wie alle Bestandteile zusammenwirken.
6. Erkennen Sie, wie Dienste aus früheren Kapiteln, wie Storage und Cosmos DB, genutzt werden?
7. Welche weiteren IoT-Lösungsbeschleuniger sind verfügbar? Passt einer von ihnen zu den Ideen für Ihre eigenen Anwendungen?

21 Serverless Computing

In diesem letzten Kapitel befassen wir uns mit der Zukunft von Serverless Computing. Wenn Sie Entwickler*in sind, mag Ihnen die Idee der Container, die wir in Kapitel 19 besprochen haben, attraktiv erschienen haben. Der Grund dafür war der deutlich niedrigere Konfigurationsbedarf für die zugrunde liegende Infrastruktur Ihrer Anwendungen. Falls dies stimmt, werden Sie großen Gefallen an den Azure Serverless-Komponenten finden! Dieses Kapitel richtet sich auch an IT-Administrator*innen, die sich fragen, welchen Aufgaben sie in Zukunft erledigen müssen, wenn es keine Server mehr gibt! *Serverless Computing* ist mehr als ein Marketingbegriff, denn viele der Server- und Infrastrukturkompetenzen bleiben auch in Zukunft relevant!

Bei Azure umfassen zwei Angebote Serverless-Computing-Funktionen: Azure Logic Apps und Azure Function Apps. In diesem Kapitel untersuchen wir, was jeder Dienst bietet und wie sie zusammenarbeiten können. Um sicherzustellen, dass Ihre Serverless Anwendungen miteinander kommunizieren und Daten weiterleiten können, gehen wir auch auf Messaging-Dienste wie Azure Event Grid, Service Bus und Event Hubs ein.

21.1 Was ist Serverless Computing?

Die Behauptung, dass Serverless Computing ohne Server auskommen kann, ist schlicht und ergreifend falsch. Ein Server irgendwo führt immer noch irgendeinen Code für Sie aus. Der Unterschied zu den Workloads der IaaS-Anwendungen wie Azure VMs und PaaS-Workloads in Web Apps besteht darin, dass Serverless-Anwendungen in der Regel in kleinere diskrete Einheiten einer Anwendung zerlegt werden. Sie führen keine einzelne, umfangreiche Anwendung aus, sondern Anwendungskomponenten, und zwar bitweise. Wenn dies für Sie nach Containern und Microservices klingt, die wir in Kapitel 19 besprochen haben, machen Sie sich keine Sorgen – Sie sind noch nicht verrückt geworden. Serverless Computing hat tatsächlich viele Überschneidungen mit diesen Themen in Bezug auf die Entwicklung Ihrer Anwendungen. Sie könnten wahrscheinlich Microservices erstellen, indem Sie die Serverless-Ansätze verwenden, die wir in diesem Kapitel betrachten werden.

Abbildung 21.1 zeigt, wie eine Anwendung in kleine Komponenten zerlegt wird, die bei einem Anbieter für Serverless Computing ausgeführt werden und kleine Ausgabeeinheiten liefern.

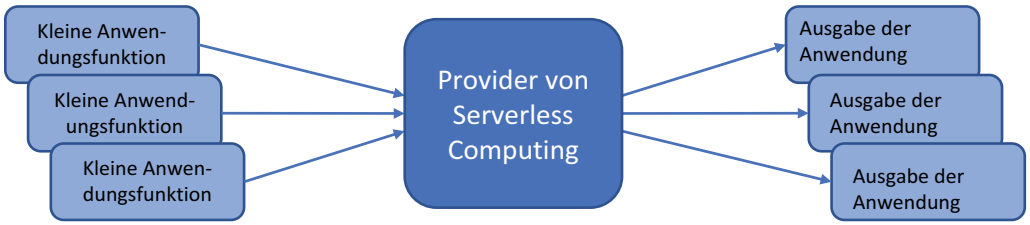


Abbildung 21.1 In einer Serverless Computing-Umgebung wird jede Anwendung in kleine, diskrete Einheiten von Anwendungskomponenten aufgeteilt. Jede Komponente wird bei einem Anbieter für Serverless Computing wie beispielsweise Azure Function Apps ausgeführt. Es wird eine Ausgabe erzeugt, die dann von anderen Serverless-Anwendungskomponenten oder anderen Azure-Diensten wie Azure IoT oder Azure Storage verwendet werden kann.

Bei Azure umfasst Serverless Computing zwei primäre Services:

- *Azure Logic Apps* – um auf bestimmte Eingaben und Auslöser zu reagieren, können Sie, ohne programmieren zu müssen, mit Logic Apps visuell Workflows erstellen, die zusätzliche Aktionen auf einfache Weise verarbeiten und generieren. Logic Apps können von Benutzenden ohne Programmier- oder IT-Infrastruktur-Kenntnisse erstellt werden. In Abbildung 21.2 ist eine einfache Übersicht über die Logic App dargestellt.

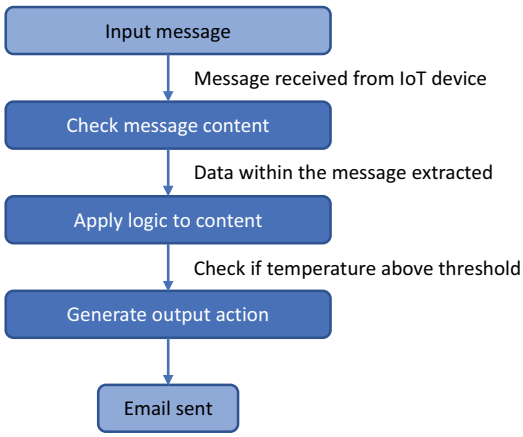


Abbildung 21.2 In einer Logic App kann eine Eingabe erfolgen, wenn ein Tweet gepostet, eine Datei hochgeladen oder eine Nachricht von einem IoT-Gerät empfangen wird. Die Logic App wendet Regeln und Filter auf die Daten an und bestimmt, ob die Nachricht die von Ihnen definierten Kriterien erfüllt. Ausgabeaktionen, wie das Erzeugen einer E-Mail, werden dann abgeschlossen. All diese Logik erfordert keine Programmier- oder Anwendungsinfrastruktur außer einem Azure-Abonnement.

Sie müssen keine Sicherheitsupdates verwalten, und es sind keine Designanforderungen an Hochverfügbarkeit oder Skalierbarkeit erforderlich. Die Azure-Plattform übernimmt diese Aufgaben automatisch. Es gibt Hunderte von vorkonfigurierten Konnektoren, mit denen sich Logic Apps mit Diensten wie Twitter, Office 365, SharePoint und Outlook integrieren lassen. Sie können auf öffentliche Tweets zu Ihrem Unternehmen oder Produkt reagieren, eine E-Mail-Benachrichtigung senden, wenn eine Datei in SharePoint hochgeladen wird, oder eine Benachrichtigung senden, wenn eine Nachricht von einem IoT-Gerät empfangen wird.

- *Azure Function Apps* – um kleine Codeblöcke auszuführen, können Sie mit Funktionsanwendungen gängige Programmiersprachen wie C#, Node.js und Python ohne zusätzliche Infrastrukturverwaltung verwenden. Ihr Code wird in einer sicheren, isolierten Umgebung ausgeführt, und die Abrechnung erfolgt nach Speicherverbrauch pro Sekunde. Abbildung 21.3 zeigt den grundlegenden Prozess für eine Function App.

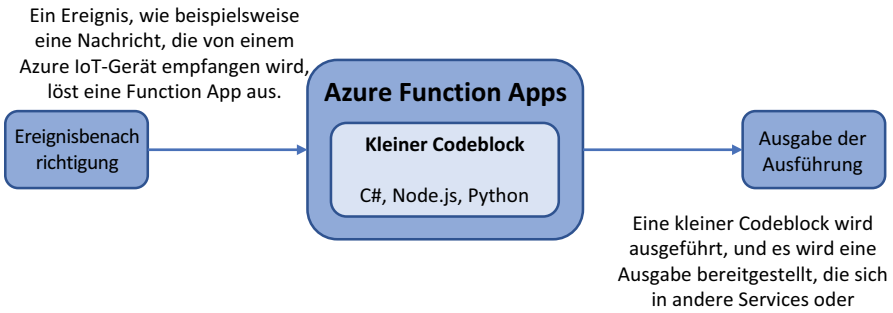


Abbildung 21.3 Wie bei einer Logic App startet eine Ereignisbenachrichtigung oder ein Auslöser in der Regel eine Azure-Funktion. Die Function App enthält eine kleine Codeeinheit, die eine bestimmte Aufgabe ausführt. Es gibt keine Infrastruktur, die konfiguriert oder gewartet werden muss. Lediglich Ihr kleiner Codeblock ist erforderlich. Nach Abschluss der Codeausführung kann die Ausgabe in einen anderen Azure-Dienst oder eine andere Anwendung integriert werden.

  Es gibt keine zu wartende VM, und es ist keine Webanwendung erforderlich. Sie müssen sich keine Sorgen um Hochverfügbarkeit oder Skalierung machen, denn der Azure Function Apps-Dienst übernimmt dies für Sie. Alles, was Sie zur Verfügung stellen müssen, ist Ihr Code. Die Azure-Plattform stellt sicher, dass immer dann, wenn Sie diesen Code ausführen müssen, Ressourcen zur Bearbeitung Ihrer Anfrage bereitgestellt werden.

Logic Apps benötigen keinen Code, sodass sie über eine größere potenzielle Benutzerbasis verfügen. Besitzer von Geschäftsanwendungen oder Finanz- und Buchhaltungsteams können beispielsweise ihre eigenen Logic Apps erstellen, ohne programmieren zu müssen. Function Apps bieten mehr Kontrolle und Flexibilität und ermöglichen es Ihnen, Ereignisse gezielt zu verarbeiten und besser in andere Anwendungskomponenten zu integrieren.

Sowohl Logic als auch Function Apps bieten Ihnen die Möglichkeit auf Auslösern basierte Aktionen durchzuführen, ohne eine Anwendungsumgebung oder Infrastruktur aufrechterhalten zu müssen. Ein Server irgendwo auf Azure führt Ihre Logic App oder -Funktion aus, doch aus Ihrer Sicht als IT-Administrator*in oder Entwickler*in handelt es sich um Serverless-Technologien.

21.2 Azure-Messaging-Plattformen

In Kapitel 12 befassten wir uns mit der Überwachung und Fehlerbehebung für Azure-Ressourcen. In Kapitel 16 wurde gezeigt, wie Sie das Azure Security Center einsetzen können, um Probleme zu erkennen und Updates zu verwalten. Beide Funktionen basieren auf Datenströmen, wie beispielsweise die Azure VM-Diagnoseerweiterung, um die Plattform über Ereignisse in der VM zu informieren. Die Diagnose- und Überwachungsplattform von Azure bieten großartige Möglichkeiten. Aber auch andere Dienste wie Web Apps, Azure Container Instances und Azure IoT Hub können die Service-Diagnostik für die zentrale Analyse streamen.

Bei Serverless Anwendungen benötigen Sie häufig eine Möglichkeit, um Nachrichten auszutauschen sowie aktuelle Anwendungsdaten und nicht nur Fehlerdiagnosen oder Statusaktualisierungen zu übermitteln. Genau dann benötigen Sie eine Messaging-Plattform.

21.2.1 Azure Event Grid

Und wenn Sie nur über bestimmte abgeschlossene Aktionen oder Aktivitäten berichten möchten? In Automatisierungs-Workflows und Serverless Computing ist die Fähigkeit, eine Aktion als Reaktion auf ein Ereignis durchzuführen, nützlich, wie in Abbildung 21.4 dargestellt.

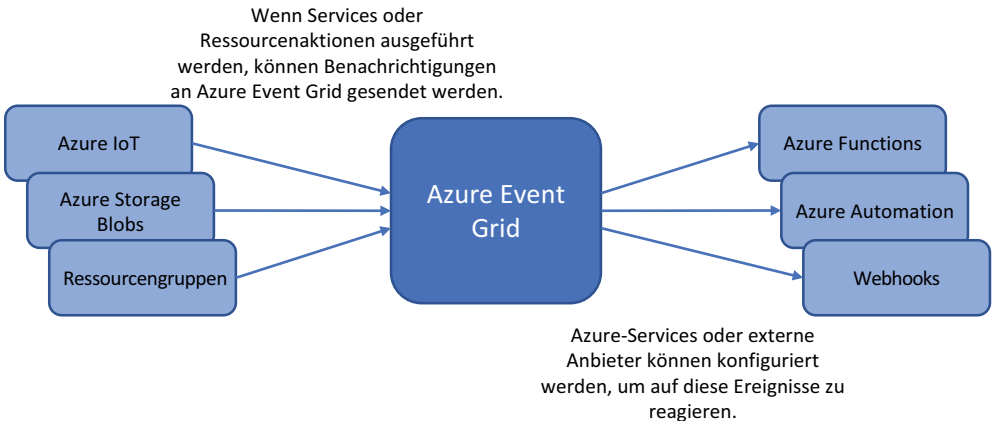


Abbildung 21.4 Azure-Dienste wie Azure IoT und Azure Storage können Benachrichtigungen an Azure Event Grid senden. Diese Benachrichtigungen erfolgen möglicherweise, wenn eine Nachricht von einem IoT-Gerät empfangen oder eine Datei in den Speicher hochgeladen wird. Mit Azure Event Grid können andere Dienste und Anbieter diese Benachrichtigungen abonnieren, um zusätzliche Aktionen als Reaktion auf Ereignisse durchzuführen.

Lassen Sie uns ein paar Szenarien untersuchen, die Sie möglicherweise in Ihrem Pizzageschäft verwenden können:

- *In einem IoT Hub empfangene Nachricht* – Ein an den IoT Hub angeschlossenes IoT-Gerät kann Daten einer Temperaturmessung in einem Ofen oder am Standort eines Lieferfahrzeugs melden. Der IoT Hub ist so konfiguriert, dass er eine Benachrichtigung an Azure Event Grid weiterleitet.

  Eine Azure-Funktion wird an die Event-Grid-Benachrichtigungen für den IoT Hub abonniert und führt eine kleine Serverless Anwendungskomponente aus, um die Informationen in Azure Cosmos DB zu protokollieren und eine E-Mail-Benachrichtigung zu senden. Sie könnten auch Logic Apps anstelle von Azure Function Apps verwenden, je nachdem, wie komplex die Antwort auf die Anwendung sein muss.

- *Auf Azure Storage hochgeladene Datei* – die Marketingabteilung kann einen Gutschein in den Speicher hochladen, um Geld bei einer Pizza-Bestellung zu sparen. Bei der Erstellung einer neuen Datei wird eine Benachrichtigung an Event Grid gesendet.

  Ein Webhook wird zum Event Grid abonniert und sendet eine Kopie des Bilds vom Speicher auf Twitter. Dieser Tweet informiert die Kunden über den Deal der Woche oder den Geldsparcoupon.

Diese Szenarien sind wirklich für reines Serverless Computing ausgelegt, aber Event Grid kann auch in traditionellere Ressourcen wie VMs und Web Apps integriert werden. So kann beispielsweise eine Ressourcengruppe konfiguriert werden, um Benachrichtigungen an Event Grid zu senden. Es gibt viele Möglichkeiten zur Erstellung einer VM, unter anderem im Portal, mit der Azure CLI oder mit einer Resource Manager-Vorlage. Sie müssen deshalb sicherstellen, dass die VM für das Update Management über das Security Center korrekt konfiguriert ist. Ein Azure Automation-Runbook könnte für Benachrichtigungen über VM-Erstellungsabläufe im Event Grid abonniert werden. Es würde dann die VM in den Update Management-Dienst einbinden und die erforderliche Sicherheitsstufe oder Anwendung installieren.

21.2.2 Azure Event Hubs und Service Bus

Event Grid kann mit vielen Azure-Ressourcen zusammenarbeiten und ist für Serverless Computing mit Logik- oder Funktionsanwendungen optimal geeignet. Aber auch Logic und Function Apps können auf der Grundlage anderer Dateneingaben, wie z. B. Event Hubs oder einem Service Bus, ausgeführt werden. Schauen wir uns die Unterschiede zwischen diesen verschiedenen Messaging-Diensten an, damit Sie sich für die richtigen Dienste entscheiden können:

- *Mit Azure Event Hubs* erhalten Sie einen Datenstrom unter anderem von IoT-Geräten oder Anwendungstelemetrie. Event Hubs bieten eine Messaging-Plattform mit niedrigen Latenzzeiten, die Millionen von Ereignissen pro Sekunde von mehreren Anbietern gleichzeitig verarbeiten kann. Bei Event Hubs handelt es sich um einen Datenspeicher statt um eine Warteschlange von Nachrichten. Der Client oder die Anwendung überprüft die Ereignisse im Hub, unabhängig von Ihrer gewünschten Häufigkeit. Dann können die Daten, die im Event Hub empfangen werden, von anderen Diensten verarbeitet werden, wie in Abbildung 21.5 gezeigt.
- *Mit Azure Service Bus* können Anwendungskomponenten Nachrichtendaten austauschen, wie beispielsweise die Speicherwarteschlangen, die wir in Kapitel 4 untersucht haben. Bei Storage Queues handelt es sich um eine ältere, einfachere Implementierung einer Messaging-Plattform in Azure. Ein *Service Bus*

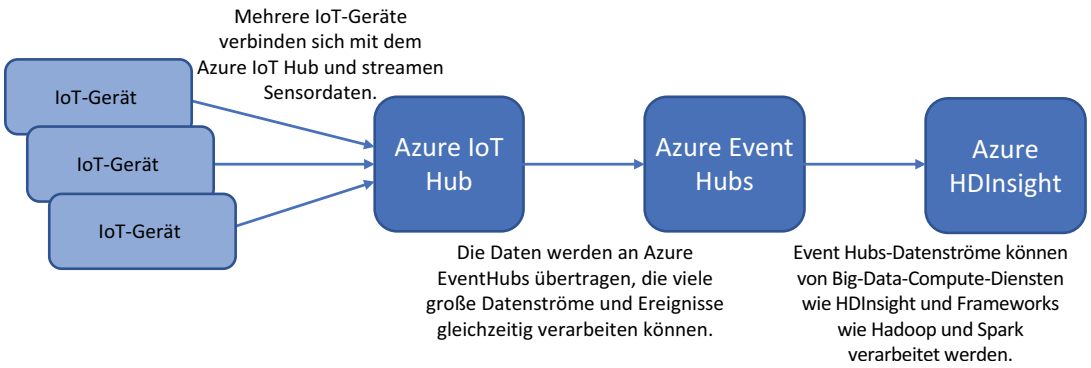


Abbildung 21.5 IoT-Geräte verbinden sich mit dem IoT Hub und können alle ihre Sensordaten streamen. Es kann Hunderte oder Tausende von angeschlossenen IoT-Geräten geben. Azure Event Hubs verarbeitet all diese separaten Datenströme und ermöglicht es Diensten wie Azure HDInsight, die Rohdaten in Hadoop oder Spark Clustern zu verarbeiten, um Berichte zu analysieren und zu generieren.

bietet erweiterte Funktionen wie garantierte Reihenfolge der Nachrichten, atomare Operationen und das Senden von Nachrichten in Stapeln. Die Abbildung 21.6 zeigt ein gängiges Szenario für einen Service Bus.

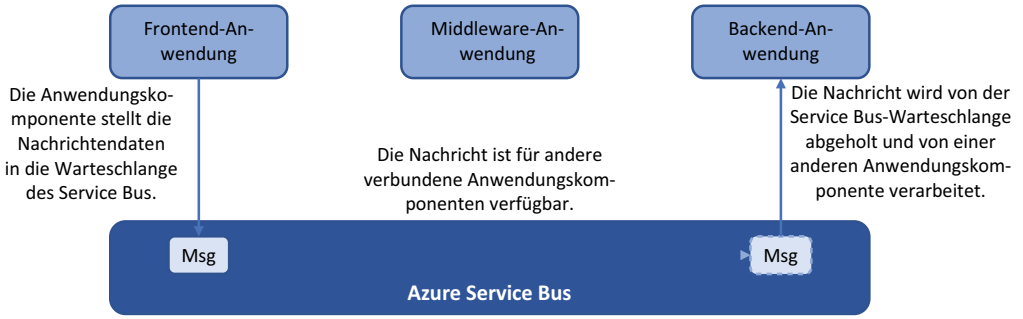


Abbildung 21.6 Nachrichten werden von Anwendungskomponenten – in diesem Beispiel einer Frontend-App – in eine Service Bus-Warteschlange gestellt. Andere Middleware- oder Backend-Anwendungen können diese Nachrichten dann aufnehmen und bei Bedarf verarbeiten. Hier nimmt eine Backend-Anwendung die Nachricht auf und verarbeitet sie. Zu den erweiterten Messaging-Funktionen zählen die Gewährleistung der Nachrichtenreihenfolge in der Warteschlange, Sperrnachrichten, Timeouts und Relais.

Welche der drei Dienste, mit denen Sie Daten zwischen Anwendungen und Diensten in Azure übertragen, empfangen und verarbeiten können, verwenden Sie und wann? Tabelle 21.1 bietet einen Überblick über die Event Grid-, die Event Hubs- und die Service Bus-Dienste.

Tabelle 21.1 Jeder Dienst ist für ein anderes Szenario konzipiert. Mit dem Event Grid können Sie auf Ereignisse reagieren, mit dem Event Hub große Datenmengen streamen und mit dem Service Bus Nachrichten zwischen Diensten und Anwendungskomponenten übertragen.

Azure service	Provides	Use case
Event Grid	Distribution of events	Perform an additional action based on an event occurrence.
Event Hubs	Data streams	Receive and transmit large volumes of concurrent data.
Service Bus	Message transmission	Provide communication between services and apps.

Azure Logic Apps und Function Apps können von allen drei Messaging-Plattformen ausgelöst werden. Erstellen wir jetzt einen Service Bus, um eine Logic App auszulösen.

21.2.3 Erstellen eines Service Bus und Integration in einen IoT Hub

In diesem Szenario verwenden Sie einen Service Bus, um Nachrichten zu senden, die von einem IoT Hub empfangen werden. Ihr simuliertes Raspberry-Pi-Gerät aus Kapitel 20 erzeugt Temperaturwerte und überträgt diese an Ihren IoT Hub. Wenn die Temperatur 30 °C übersteigt, ist ein weiteres Datensegment in der Nachricht des IoT-Geräts enthalten: `temperature-Alert = true`. Abbildung 21.7 zeigt, wie Sie einen IoT Hub in den Service Bus integrieren können, um Nachrichten mit diesem Temperaturalarm zu verarbeiten.

Abbildung 21.7 Wenn Ihr simuliertes Raspberry Pi IoT-Gerät Nachrichtendaten sendet, erzeugt eine Temperaturmessung von 30 °C oder mehr einen Alarm. Nachrichten mit dieser Warnung werden in einem Service Bus platziert. Diese Nachrichten können dann genutzt werden, um Logic Apps auszulösen.

Jetzt testen

Um einen Service Bus zu erstellen, führen Sie die folgenden Schritte aus:

1. Öffnen Sie das Azure Portal, und klicken Sie im Menü oben links auf „Create a Resource“ (Ressource erstellen).
2. Suchen und wählen Sie den Service Bus, und wählen Sie dann „Create“ (Erstellen).
3. Geben Sie einen Namen ein, wie z. B. `azuremol`, und wählen Sie dann die Grundpreisstufe.
4. Erstellen Sie eine neue Ressourcengruppe, und geben Sie einen Namen an, z. B. `azuremol-chapter21`. Stellen Sie sicher, dass der Standort der gleiche ist wie für die Ressourcen, die in Kapitel 20 erstellt wurden, wie z. B. USA Osten. Die Interaktion zwischen einer Service Bus-Warteschlange, einer Logic App und einer Function App kann zu Problemen führen, wenn Sie nicht mit Ihren Standorten übereinstimmen.
5. Akzeptieren Sie die anderen Standardwerte, und erstellen Sie den Service Bus.
6. Nachdem die Ressource erstellt wurde, wählen Sie Ihre Ressourcengruppe aus. Wählen Sie dann den Service Bus, den Sie in Schritt 5 erstellt haben.
7. Wählen Sie die Warteschlange aus, und geben Sie einen Namen z. B. `azuremol` ein.
8. Lassen Sie alle anderen Standardwerte unverändert, und wählen Sie „Create“ (Erstellen).

Wie konfigurieren Sie einen IoT Hub, um ihn bei der Erstellung eines Service Bus und einer Warteschlange zu verwenden? Im IoT Hub definieren Sie *Endpunkte* als Ziel für Nachrichten, die von IoT-Geräten empfangen werden. Im IoT Hub existiert ein Standard-Endpunkt für alle Nachrichten, die die definierten Kriterien nicht erfüllen. Sie können den Service Bus als Endpunkt für den Empfang von Nachrichten konfigurieren. Anschließend wird eine *Route* mit Kriterien definiert, für die Nachrichten an einen Endpunkt geleitet werden sollen. In diesem Beispiel erfordert diese Route jede Nachricht, die `temperatureAlert = true` im Nachrichtentext enthält, um zum Endpunkt des Service Bus geleitet zu werden, wie in Abbildung 21.8 gezeigt.

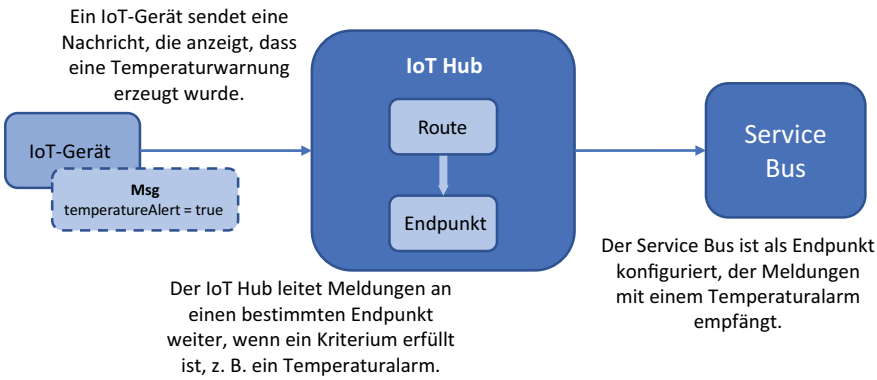


Abbildung 21.8 Da Nachrichten von IoT-Geräten an einen IoT Hub übertragen werden, können sie nach den von Ihnen definierten Kriterien an bestimmte Endpunkte weitergeleitet werden. Nachrichten mit einen Temperaturalarm im Nachrichten-Body können an einen Endpunkt weitergeleitet werden, der die Service Bus-Warteschlange verwendet. Die in der Warteschlange des Service Bus platzierten Nachrichten mit einem Temperaturalarm können dann verwendet werden, um unter anderem Azure Logic Apps oder Function Apps auszulösen.

Jetzt testen

Führen Sie die folgenden Schritte aus, um einen IoT Hub so zu konfigurieren, dass Temperaturwarnmeldungen an den Service Bus weitergeleitet werden.

1. Wählen Sie Ihre Ressourcengruppe aus Kapitel 20 aus, wie z. B. `azuremolchapter20`, und wählen Sie dann den IoT Hub.
2. Wählen Sie unter „Messaging“ in der linken Navigationsleiste die Option „Message Routing“ (Nachrichtenweiterleitung), und fügen Sie einen individuellen Endpunkt für eine Service-Bus-Warteschlange hinzu.
3. Geben Sie einen Endpunktnamen ein, z. B. `azuremol`.
4. Wählen Sie den Warteschlangen-Namensraum für Ihren Service Bus, wie z. B. `azuremol`, und wählen Sie dann Ihre eigentliche Warteschlange.
5. Erstellen Sie eine Route, um Nachrichten an diesen Endpunkt zu weiterleiten. Wählen Sie im Abschnitt „Message Routing“ (Nachrichtenweiterleitung) in der Navigationsleiste auf der linken Seite „Routes“ (Routen) aus, und fügen Sie eine neue Route hinzu.
6. Geben Sie einen Namen ein, wie z. B. `temperatureAlert`.
7. Wählen Sie den Service Bus-Endpunkt, den Sie im vorherigen Schritt erstellt haben, z. B. `azuremol`.
8. Geben Sie für die Routing-Abfrage Folgendes ein:

```
temperatureAlert = "true"
```

9. Wenn Sie bereit sind, speichern Sie die Route.

Jetzt haben Sie ein simuliertes Raspberry Pi-Gerät, das Daten an den IoT Hub sendet, und eine Route, um Nachrichten mit einem Temperaturalarm in eine Warteschlange für Service Bus-Nachrichten zu stellen. Sie verfügen noch über keine Anwendung, daher können Sie mit den Daten in der Warteschlange des Service Busses noch nichts unternehmen. Was könnten Sie mit einer Temperaturwarnung machen? Das

Versenden einer E-Mail-Benachrichtigung ist ein häufiges Beispiel. Lassen Sie uns deshalb darauf eingehen, wie eine Logic App ausgelöst werden kann, wenn eine Nachricht in der Warteschlange eines Service Bus platziert wird.

21.3 Erstellen einer Azure Logic App

Wie wir bereits in Abschnitt 21.1 über Logic Apps besprochen haben, kann eine von einer Service Bus-Warteschlange empfangene Nachricht als Auslöser verwendet werden, um den Ausführungsprozess zu starten. Sie verwenden den IoT Hub, um die von IoT-Geräten empfangenen Nachrichten zu verarbeiten und nur zu den Endpunktnachrichten der Service Bus-Warteschlange zu leiten, die `temperatureAlert = true` im Nachrichtentext enthalten. Bei diesem Ansatz wird Ihre Logic App nur dann ausgeführt, wenn ein Temperaturalarm ausgelöst wird.

Abbildung 21.9 zeigt, was Ihre Logic App macht. Wenn eine Nachricht in die Warteschlange des Service Bus gestellt wird, wird die Logic App ausgeführt und sendet eine E-Mail-Benachrichtigung.

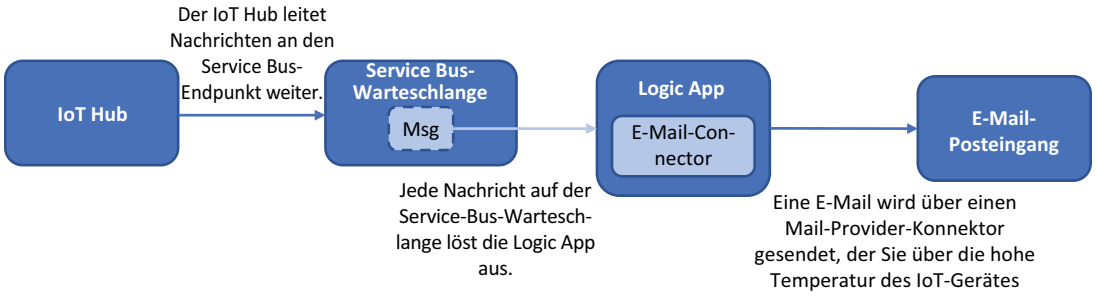


Abbildung 21.9 Jede Nachricht, die vom IoT Hub in der Service Bus-Warteschlange empfangen wird, löst die Logic App aus. Wenn die Logic App ausgeführt wird, sendet sie eine E-Mail-Benachrichtigung über einen definierten E-Mail-Anbieter.

Jetzt testen

Führen Sie die folgenden Schritte aus, um Ihre Logic App zu erstellen:

1. Klicken Sie oben links im Menü des Azure Portals auf Ressource erstellen.
2. Suchen und wählen Sie die Logic App, und wählen Sie dann „Create“ (Erstellen).
3. Geben Sie einen Namen an, wie z. B. `azuremol`, und wählen Sie Ihre Ressourcengruppe aus, wie z. B. `azuremolchapter21`. Auch hier wählen Sie den gleichen Ort wie für Ihre anderen IoT-Ressourcen aus Kapitel 20.
4. Akzeptieren Sie die anderen Standardwerte, und wählen Sie „Create“ (Erstellen).
5. Nach dem Erstellen der Ressource wählen Sie Ihre Ressourcengruppe aus, und öffnen Sie dann die Logic App. Wählen Sie für die Option „Add Common Triggers“ (Allgemeine Trigger hinzufügen) „When a message is received in a Service Bus queue“ (Wenn eine Nachricht in einer Service Bus-Warteschlange empfangen wurde).
6. Geben Sie einen Namen an, z. B. `azuremol`. Wählen Sie dann Ihre Service Bus-Warteschlange aus, z. B. azuremol.

7 Wählen Sie die angegebene Standard-Service Bus-Richtlinie, z. B. RootManageSharedAccess-Key, und erstellen Sie die Verbindung.

8 Wählen Sie „Continue“ (Weiter), und wählen Sie dann den Namen Ihrer Service Bus-Warteschlange wie z. B. `azuremol` aus.

9 Übernehmen Sie die Standardwerte, z. B. die Häufigkeit, um nach Nachrichten zu suchen.

10 Fügen Sie der Logic App einen neuen Schritt hinzu.

11 Um eine Aktion hinzuzufügen, suchen Sie nach dem, was Sie tun möchten. Suchen Sie in dieser Übung nach *E-Mails*. Wählen Sie Ihren Anbieter z. B. Gmail aus – Eine E-Mail senden, Outlook.com – Eine E-Mail senden oder SMTP – Eine E-Mail senden, wie in Abbildung 21.10 dargestellt.

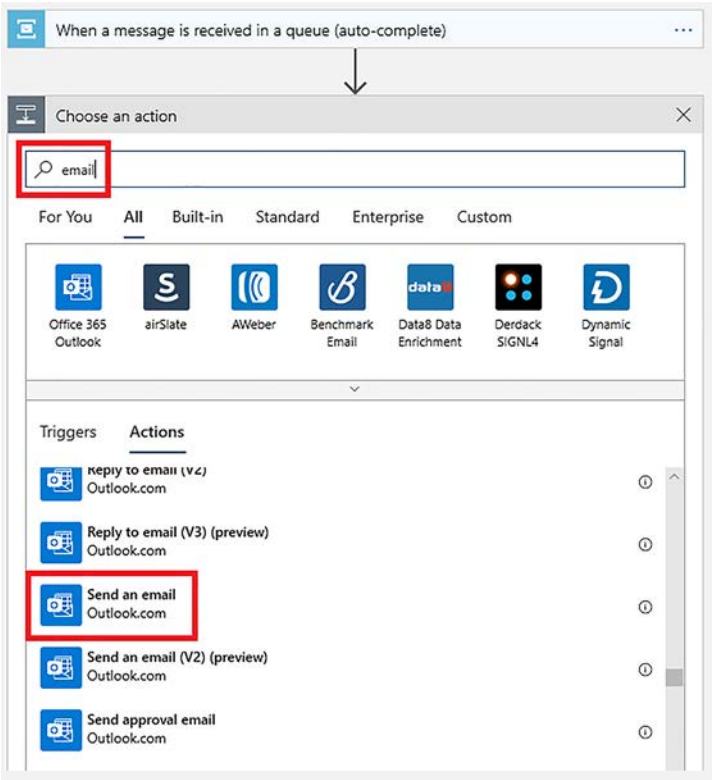


Abbildung 21.10 Suchen und wählen Sie Ihren aktuellen E-Mail-Anbieter, z. B. Gmail oder Outlook.com. Sie können auch SMTP wählen – Eine E-Mail senden, um einen anderen Anbieter manuell zu konfigurieren.

12 Melden Sie sich bei Ihrem E-Mail-Anbieter an, um E-Mail-Routing zu genehmigen und zu bestätigen, dass Sie Logic Apps Berechtigungen zum Versenden von E-Mails gewähren möchten.

13 Geben Sie eine E-Mail-Adresse des Empfängers an, um E-Mails zu erhalten, einen Betreff, wie z. B. `Temperaturalarm`; und einen Nachrichtentext, wie z. B. `Hohe Temperatur auf dem IoT-Gerät erkannt.`

14 Speichern Sie die Logic App.

Legen wir hier eine Pause ein und schauen uns an, was Sie wie in Abbildung 21.11 dargestellt in den letzten Übungen erstellt haben. Dieses grundlegende Serverless Anwendungsdesign enthält keine Steuerelemente, die die Anzahl der zu sendenden Nachrichten begrenzen. In der Logic App können Sie festlegen, dass Sie nur maximal fünf E-Mail-Benachrichtigungen senden und dann 30 Minuten warten, bevor Sie Weitere senden. Im Rahmen Ihres Anwendungsdesigns sollten Sie überlegen, wie Sie über solche Situationen informiert werden möchten. Sie können die Logic App auch so konfigurieren, dass sie die Nachrichtendaten aus der Service Bus-Warteschlange einliest und den Zeitstempel der IoT-Gerätenachricht sowie die aktuell aufgezeichnete Temperatur einbezieht. Wie das geht, besprechen wir in der nächsten Übung.

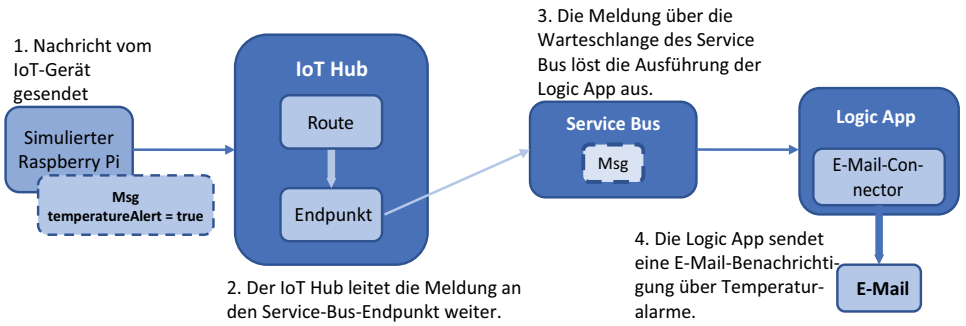


Abbildung 21.11 Das simulierte Raspberry Pi-Gerät sendet alle zwei Sekunden eine Nachricht mit Temperatursensorwerten an den IoT Hub. Liegt die Temperatur über 30 °C, wird ein Temperaturalarm aufgezeichnet. Der IoT Hub leitet alle Nachrichten mit einem Temperaturalarm an eine Service Bus-Warteschlange weiter. Nachrichten in dieser Warteschlange lösen die Ausführung einer Azure Logic App aus. Die Logic App ist mit einem E-Mail-Anbieter wie Outlook oder Gmail verbunden und sendet eine E-Mail-Benachrichtigung über die Temperaturwarnung vom IoT-Gerät.

Betrachten wir diese grundlegende Serverless Anwendung in Aktion.

Jetzt testen

Führen Sie die folgenden Schritte aus, um Ihr simuliertes Raspberry Pi-Gerät auszuführen und Ihre Logic App zu testen.

1. Öffnen Sie einen Webbrowser für das simulierte Raspberry Pi-Gerät aus Kapitel 20 (https://azure-samples.github.io/raspberry-pi-web-simulator).
2. Vergewissern Sie sich, dass Ihre IoT-Hub-Verbindungszeichenfolge immer noch im Codefenster eingetragen ist, das Sie in Kapitel 20 konfiguriert haben.
3. Führen Sie die App aus.

   Alle zwei Sekunden werden simulierte Temperatur- und Feuchtigkeitssensoren erzeugt und eine Nachricht an den IoT Hub gesendet. Ein simulierter Temperaturmesswert von 30 °C wird möglicherweise erst nach einigen Meldungen erzeugt und im Ausgabefenster angezeigt.

Der IoT Hub leitet alle Nachrichten mit `temperatureAlert: true` an den Service Bus-Endpunkt weiter. Da diese Nachrichten in der Service Bus-Warteschlange platziert werden, nimmt die Logic App sie auf und sendet eine E-Mail über den definierten Anbieter. Sie erhalten dann eine E-Mail, die Sie über die Messung der hohen Temperaturwerte informiert. Dieser Vorgang sollte insgesamt nur wenige Sekunden dauern.

4. Das simulierte Raspberry Pi-Gerät erzeugt alle zwei Sekunden Nachrichten. Halten Sie die App deshalb an, es sei denn, Sie mögen viele E-Mail-Benachrichtigungen!

Die Nachricht der E-Mail-Alarme enthält nicht viele Informationen. Ihre Logic App extrahiert nicht die Nachrichteninhalte aus dem Service Bus und formatiert die Informationen nicht. Es wäre großartig, wenn die Alarm-E-Mail den Namen des IoT-Geräts oder die aufgezeichnete Temperatur enthalten könnte. Wie können Sie jede Nachricht verarbeiten und eine Analyse durchführen? Wie sieht es mit dem anderen Azure Serverless-Dienst aus – Azure Function Apps?

21.4 Erstellen einer Azure Function App zur Analyse von IoT-Gerätedaten

Um Ihre aktuelle Serverless Anwendung zu erweitern, können Sie aus Ihrer Logic App heraus eine Azure Function App auslösen. Messdaten aus dem Service Bus können an eine Function App zur Analyse der aufgezeichneten Temperatur gesendet werden. Die von der Logic App gesendete E-Mail-Benachrichtigung kann dann Informationen zum IoT-Gerätenamen und die aufgezeichnete Temperatur enthalten. Die Interaktion zwischen der Logic und der Function App ist in Abbildung 21.12 dargestellt.

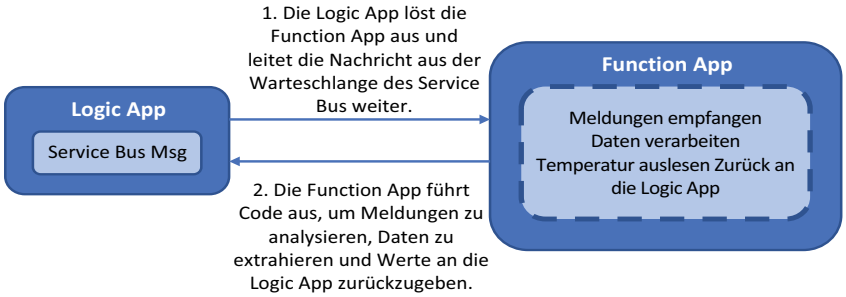


Abbildung 21.12 Die Logic App löst die Function App aus. Die auf der Warteschlange des Service Busses eingegangene Nachricht wird an die Funktion übergeben. Der Code in der Function App parst die Nachricht, extrahiert die Temperatur und gibt diesen Wert an die Logic App zurück. Es dauert einige Millisekunden, bis die Function App diesen Code ausführt, sodass die Kosten für die Ausführung dieser Rechenaufgaben Bruchteile eines Cent betragen.

Jetzt testen

Führen Sie die folgenden Schritte durch, um eine Function App zu erstellen und sie von der Logic App aus auszulösen:

1. Klicken Sie oben links im Menü des Azure Portals auf „Create a Resource“ (Ressource erstellen).
2. Suchen und wählen Sie die Function App, und wählen Sie dann „Create“ (Erstellen).
3. Wählen Sie Ihre Ressourcengruppe aus, z. B. `azuremolchapter21`, und geben Sie einen Namen an, wie z. B. `azuremol`. Sie müssen sich in der gleichen Region wie Ihre vorherigen Ressourcen befinden.

   Sie möchten Code veröffentlichen, aber beachten Sie, dass Sie auch ein Docker-Container-Image (Kapitel 19) veröffentlichen können. Sie müssen nicht einmal eine Container-Instanz oder eine zusätzliche Infrastruktur erstellen. Ein kurzlebiger Container würde bei Bedarf ausgeführt und dann angehalten.
4. Wählen Sie für diese grundlegende Anwendung die Node. js-Laufzeit, da wir ein einfaches JavaScript verwenden.
5. Sie haben drei Optionen für den Hosting-Plan. Mit einem *Verbrauchsplan* können Sie pro Ausführung bezahlen, und die benötigten Ressourcen werden der Laufzeit dynamisch zugewiesen. Für konsistente, produktionsreife Anwendungen können Sie einen *Premium-* oder *dedizierten Hostingplan* mit festeren, vorhersehbareren Kosten verwenden. Premium-Pläne bieten zusätzliche Funktionen, wie z. B. die Sicherung der Konnektivität mit einem definierten Satz von virtuellen Azure-Netzwerken und eine Instanz, die einige Verzögerungen in einem Cold-Start-Szenario für Ihre App verhindern kann. Wählen Sie für diese Übung einen Verbrauchstarif.
6. Akzeptieren Sie die anderen Standardwerte, um ein benanntes Speicherkonto und Application Insights zu erstellen, und wählen Sie dann „Review + Create“ (Überprüfen + Erstellen).
7. Wenn Sie bereit sind, erstellen Sie die Function App. Die Erstellung der Function App kann ein bis zwei Minuten in Anspruch nehmen.
8. Nach dem Erstellen der Ressource wählen Sie Ihre Ressourcengruppe aus, öffnen Sie Ihre Logic App aus der vorherigen Übung, und wählen Sie „Edit“ (Bearbeiten).
9. Wählen Sie im Logic Apps-Designer einen neuen Schritt aus.
10. Suchen und wählen Sie „Azure Functions“, und wählen Sie dann die in den vorherigen Schritten erstellte Funktion, wie z. B. `azuremol`. Wählen Sie dann „Create New Function“ (Neue Funktion erstellen).
11. Geben Sie einen Funktionsnamen an, z. B. `analyzeTemperature`.
12. Löschen Sie einen bestehenden Code, ersetzen Sie ihn durch den Code aus den folgenden Verzeichnissen, und wählen Sie dann „Create“ (Erstellen).

    Dieser Code ist auch im GitHub-Repository unter w verfügbar.

Auflistung 21.1 `analyzeTemperature` JavaScript-Code für eine Function App

Erstellt die Funktion. Jede JavaScript-Function App beginnt mit dem Export einer Funktion mit einem Kontextobjekt. Dieses Kontextobjekt wird verwendet, um Daten in beiden Richtungen zu übermitteln.

Liest Nachrichteninhalte aus Service Bus ein

Erstellt ein JSON-Objekt mit dekodierten Service Bus-Nachrichten

Extrahiert die aufgezeichnete Temperatur aus dem IoT-Gerät

Dekodiert aus base64

Erstellt eine Antwort zum Zurücksenden an die Logic App

Gibt die Temperatur im Konsolenprotokoll aus

```
module.exports = function (context, req) {
    var buffer = new Buffer(req.body.ContentData, 'base64')
    var decodedString = buffer.toString();
    var objects = JSON.parse(decodedString);
    var temperature = objects["temperature"];
    context.res = {
        body: {
            analysis: "Recorded temperature was  " + temperature + "!"
        }
    };
    context.log("Recorded temperature was " + temperature);
    context.done();
};
```

Beendet die Funktion. Jede JavaScript-Function App muss mit einem Aufruf von context.done enden, der der Function App mitteilt, dass Ihr Code fertiggestellt ist.

13 Aktivieren Sie im Logic App Designer für Ihren Funktionsschritt das Textfeld „Request Body“ (Anfragetext), und wählen Sie dann Service Bus-Nachricht aus der Liste mit dem dynamischen Inhalt auf der rechten Seite.

14 Nutzen Sie die Drag-and-Drop-Funktion im Logic App Designer, um die Schritte neu anzuordnen, damit die Aktion „Send an Email“ (Senden einer E-Mail) unterhalb des Schritts der Function App `analyzeTemperature` eingeordnet wird, wie in Abbildung 21.13 dargestellt.

15 Wählen Sie die Aktion „Send an Email“ (Senden einer E-Mail), und wählen Sie dann das Textfeld für den Textkörper der E-Mail-Nachricht aus.

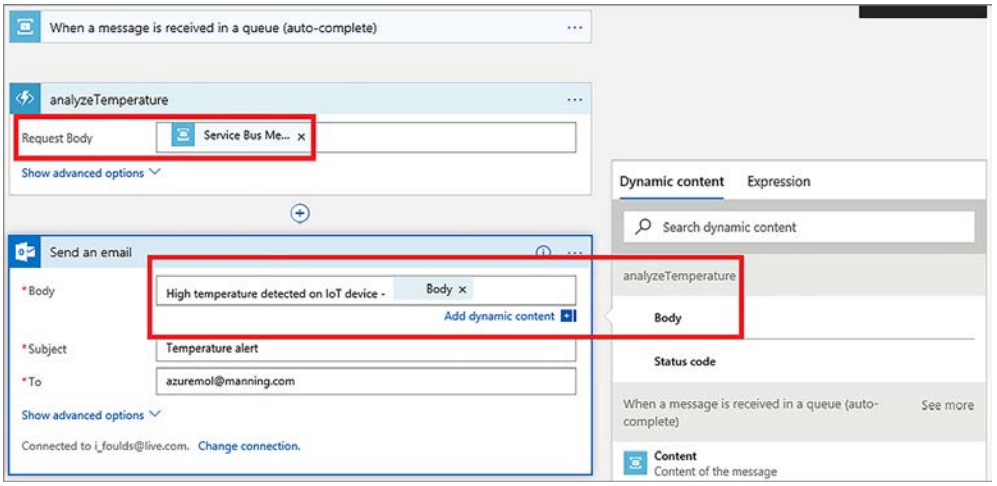


Abbildung 21.13 Ziehen Sie die Aktion „Send an Email“ (Senden einer E-Mail) unter die Funktion `analyzeTemperature`. Wählen Sie das Ende der Nachrichten-Textkörpers aus, danach wird das Dialogfeld "Dynamischer Inhalt" angezeigt. Um den von der Function App berechneten Temperaturwert einzufügen, wählen Sie den Textkörper aus der Funktion `analyzeTemperature`.

16 Wählen Sie von der Funktion `analyzeTemperature` aus die Reaktion des Textkörpers aus, wie in Abbildung 21.13 dargestellt.

17 Wählen Sie „Save“ (Speichern) im Logic App Designer.

Ihre Serverless Anwendung verfügt über viele mobile Bestandteile. Schauen wir uns jetzt an, was Sie bis jetzt erstellt haben. Danach können Sie das simulierte Raspberry Pi IoT-Gerät ausführen, um E-Mail-Benachrichtigungen mit dem von der Function App berechneten Temperaturwert zu generieren. Abbildung 21.14 enthält einen Überblick über alle Komponenten, die derzeit in der Serverless Anwendung verwendet werden.

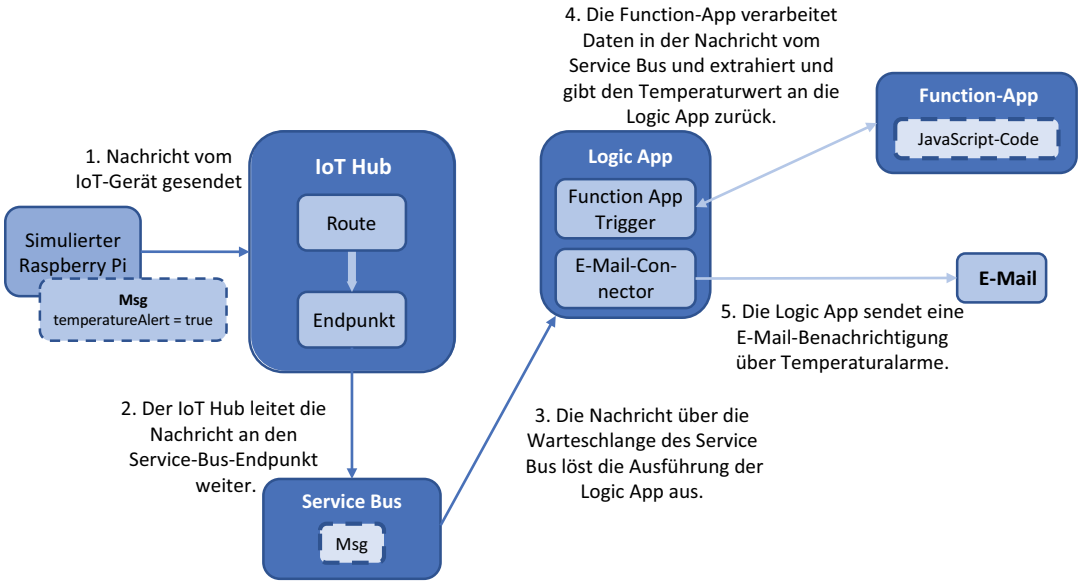


Abbildung 21.14 Beim Empfang von Nachrichten vom simulierten Raspberry Pi-Gerät werden alle Nachrichten mit einem Temperaturalarm an den Endpunkt der Service Bus-Warteschlange weitergeleitet. Nachrichten in der Service Bus-Warteschlange lösen eine Logic App aus, die die Nachricht an eine Function App weiterleitet. Eine JavaScript-Funktion analysiert den Temperaturwert und gibt ihn an die Logic App zurück. Diese sendet dann eine E-Mail-Benachrichtigung, die die von einem Sensor auf dem IoT-Gerät aufgezeichnete Temperatur enthält.

18 Öffnen Sie Ihr simuliertes Raspberry Pi-Gerät in einem Webbrowser, und führen Sie die Anwendung aus. Bei jeder Generierung des Temperaturalarms löst die Logic App die Function App aus, um die Temperaturdaten aus dem Nachrichtentext zu extrahieren und in die E-Mail-Benachrichtigung aufzunehmen. Es kann einige Augenblicke dauern, bis ein Temperaturmesswert mehr als 30 °C beträgt, und die Meldung mit einem Temperaturalarm gekennzeichnet wird. Wenn diese Warnung eingestellt ist und die Nachricht verarbeitet wird, erhalten Sie eine E-Mail-Benachrichtigung, die Ihnen den Wert dieser Temperatur mitteilt.

Atmen Sie tief durch, und klopfen Sie sich auf die Schulter. Denn es war eine Menge Arbeit für eine Mittagspause!

Authentifizierungsfehler von Ihrer Logic App zur Function App

Sie können den Ausführungsverlauf im Übersichtsfenster ihrer Logic App im Azure Portal sehen. Wenn Sie viele wiederholte Fehler erhalten, wählen Sie einen der Fehler aus, um mehr darüber zu erfahren, woran dies liegt.

Ein häufiges Problem ist, dass die Logic App nicht automatisch dazu autorisiert ist, mit der Function-App zu kommunizieren. Die erneute Bereitstellung der Logic App behebt normalerweise diesen Fehler, aber die eigentliche Lösung besteht wahrscheinlich darin, den sogenannten *Funktionsschlüssel* in den Header Ihrer Logic App einzufügen.

Um diesen Schlüssel zu erhalten, wählen Sie Ihre Function App aus, und wählen Sie dann die von Ihnen erstellte Funktion, z. B. `analyzeTemperature`. Unter der Option zum Verwalten kann der Standard-Funktionsschlüssel angezeigt und kopiert werden. Kopieren Sie diesen Schlüssel, gehen Sie zurück zu Ihrer Logic App, und öffnen Sie den Designer.

Fügen Sie in der Funktion `analyzeTemperature` einen Parameter hinzu und dann einen Header. Sie möchten zu Beginn des Aufrufs eine kleine Information an die Function App senden, die den Schlüssel sendet. Die Eingabe des Schlüsselpaares ist etwas umständlich, aber geben Sie `x-functions-key` für den Schlüssel ein, und fügen Sie dann Ihren eigentlichen Funktionsschlüssel als Wert ein.

Es dauert einige Augenblicke, um die Logic App/Function App-Integration zu aktualisieren. Danach sollte der Verlauf der Logic App zeigen, dass die Ereignisse korrekt funktionieren, und die E-Mail-Benachrichtigungen sollten zugestellt werden.

21.5 Machen Sie mit dem Lernen weiter

In diesem Kapitel wurden viele neue Konzepte vorgestellt. In der Tat haben Sie in den letzten Kapiteln viele neue Ideen und Technologien kennengelernt! Machen Sie sich keine Sorgen, wenn Sie Schwierigkeiten bei der Implementierung all dieser Azure-Dienste wie Container, KI und ML und Serverless Computing haben sollten. In diesen Kapiteln wollten wir Ihnen die Möglichkeiten in Azure demonstrieren, damit Sie sich nicht auf die Durchführung von „Lift and Schift"-Aktionen von veralteten Anwendungen beschränken müssen. Wenn Sie mit dem Erstellen und Ausführen von Anwendungen in Azure beginnen, sollten Sie die Gelegenheit nutzen, Anwendungen zu modernisieren und Verwaltungs- oder Bereitstellungs-Workflows zu überprüfen. Azure bietet viele Dienste, die den Anwendungslebenszyklus vereinfachen und beschleunigen. Sie können deshalb weitgehend unabhängig von den aktuell ausgeführten VMs handeln und müssen sich nicht an die Präferenzen Ihres Unternehmens halten.

Azure bietet viele neue, großartige Dienste an, die aber alle größtenteils auf den in Teil 1 dieses Buchs beschriebenen Kerninfrastrukturkomponenten aufbauen. Entwickler*innen können mit der Verwendung der neuesten Anwendungsdesign-Ansätze mit Kubernetes oder Serverless Computing beginnen. Administratoren können ihr Wissen zum On-Premises-Rechenzentrum mit Grundlagen von Cloud Computing und Troubleshooting-Techniken auf neuen Wegen einsetzen. Azure unterstützt Ihre Mitarbeiter*innen und das Wachstum Ihres Unternehmens.

In Kapitel 1 war ich offen und ehrlich, als ich sagte, dass ich nicht auf jeden Dienst auf Azure eingehen werde. Es gibt noch viel mehr Azure-Dienste, über die wir mehr erfahren müssen. Zu den Diensten, mit denen wir uns in diesem Buch befasst haben, gibt es auch viele weitere tiefergehende Informationen. Ich hoffe, dass mindestens ein paar Bereiche für Sie dabei waren, die Sie interessieren und motivieren, mehr zu entdecken. Meine Favoriten sind Virtual Machine Scale Sets, Cosmos DB und Azure Kubernetes Service.

21.5.1 Ergänzende Lernmaterialien

Ich bin zwar voreingenommen, aber ein großartiger Ort, um Azure besser kennenzulernen, ist https://docs.microsoft.com/azure. Diese Webseite bietet die wichtigsten Dokumente zum Azure-Dienst, Architekturleitfäden, Referenz- und SDK-Ressourcen sowie Beispiele. Jeder Azure-Dienst verfügt über sein eigenes Angebot von Schnellstart-Leitfäden, Tutorials und Beispielen, zusammen mit konzeptionellen Informationen und individuellen Anleitungen.

Wenn Sie es wirklich ernst meinen, können Sie Zertifizierungsoptionen untersuchen. Individuelle Prüfungen umfassen die Module *Microsoft Azure Administrator (AZ-104)*, *Microsoft Azure Architect Technologies and Design (AZ-303 und AZ-304)* und *Microsoft Azure Security Technologies (AZ-500)*. Dieses Buch und die Übungen, die Sie absolviert haben, decken einen Großteil der Bereiche ab, in denen diese Prüfungen Ihr Wissen testen. Sie müssen aber einige zusätzliche Bereiche von Azure AD durchgehen und Best Practices anwenden, bevor Sie die Prüfungen ablegen. Die Microsoft-Website unter https://docs.microsoft.com/learn bietet einige zusätzliche Lernpfade für die verschiedenen Azure-Zertifizierungsoptionen, die Sie bei der Vorbereitung unterstützen.

21.5.2 GitHub-Ressourcen

In diesem Buch haben Sie Codebeispiele, Vorlagen und Beispielanwendungen von https://github.com/fouldsy/azure-mol-samples-2nd-edition verwendet. Diese Beispiele sollten aktualisiert werden, wenn neue Versionen des Azure CLI veröffentlicht werden. Das GitHub-Repository enthält auch PowerShell-Beispiele und Vorlagen für alle Übungen. Dieses Buch befasst sich hauptsächlich mit dem Azure CLI in der Azure Cloud Shell. Sie können aber auch gerne jede Übung in PowerShell oder einer Vorlage testen.

Wenn Sie Probleme mit den Beispielen haben, erstellen Sie ein Issue in GitHub unter https://github.com/fouldsy/azure-mol-samples-2nd-edition/issues. Azure entwickelt sich schnell weiter. Ich möchte deshalb sicherstellen, dass Sie immer anhand neuester praktischer Beispiele lernen. Ich freue mich aber auch auf Ihre Vorschläge und Anregungen! Alle Azure-Dokumente aus https://docs.microsoft.com/azure sind mit Feedback-, Problemmeldungs- und Änderungsfunktionen ausgestattet. Auf diese Weise können Sie, wenn Sie den Rest des Angebots in Azure kennenlernen, sich gerne beteiligen und andere bei Ihrem Lern- und Entwicklungsprozess unterstützen.

21.5.3 Ein abschließender Gedanke

Holen Sie tief Luft und denken Sie dran, dass Wandel heutzutage zur Normalität dazugehört. Neue Funktionen und Dienste werden fast täglich veröffentlicht. Azure kann, wie alle wichtigsten Cloud Computing-Anbieter, immer wieder Änderungen unterliegen. Das Benutzungserlebnis kann sich deshalb bei Ihrer nächsten Anwendung etwas unterscheiden – Änderungen im Stundentakt sind hier keine Ausnahme. Mit den grundlegenden Kernkompetenzen und dem Verständnis, die Sie hoffentlich in diesem Buch erworben haben, können Sie sich an diese Änderungen anpassen und sich mit all den neuen Möglichkeiten von Azure weiterentwickeln. Es gibt immer etwas Neues zu lernen, und ich würde gerne von Ihren Erfolgen bei der Entwicklung und Ausführung von Anwendungen in Azure erfahren!

Index

Symbole

&& Buchstabe 27
$ResourceGroups-Objekt 276
$servicePrincipalConnection-Objekt 275

A

Abhängigkeiten 82
Absturzabbilder 180
access_token-Variable 226
ACI (Azure Container-Instanz) 284, 290, 292
ACR (Azure Container-Register) 293
Administrator*in für den Benutzungszugriff, Rolle 78
Agents 180
  Anzeigen von Informationen zu 294
  Ausführen von Websites auf Kubernetes 295–297
  Erstellung von Clustern mit 294–295
AKS (Azure Kubernetes Service) 284, 293–297
  Azure Security Center-NSGs 234
    OMS (Operations Management Suite) 243
    Überprüfen und Anwenden von Updates 245–249
  Azure Update Management 241–249
  JIT (Just-in-Time) 237–241, 249
Aktualisierungen 234–249
Alias-Datensätze 160
AllowAzureLoadBalancerInBound-Regel 67
AllowVnetInBound-Regel 67
Amazon Web Services (AWS) 191
Anmeldeinformationen 270
Anwenden und Autokorrektur Modus, DSC 279
Anwenden- und Überwachen-Modus, DSC 279
Anwendungen mit Lastausgleich 106–123
Anwendungssicherheitsgruppe 185
Anycast Netzwerkadressierung 160
APIs (Anwendungsprogrammierschnittstellen) 31
App Service-Pläne 35–38
App Service-Umgebungen 36
Application Gateway 107–108
  Function Apps 328–331
  Lebenszyklen von 76–77
  Logic Apps 325–328
  Service-Pläne 38
Apps
Apps. Siehe auch Azure Web Apps
APT (Advanced Packing Tool) 27
Assets, in Azure Automation 272–274
Auf 0 basierende Nummerierung 99
Aufheben von Datenträgeroptionen 199
  Erstellen von Traffic Manager-Profilen 164–166
  Globale Verteilung des Datenverkehrs auf die nächste Instanz 167–173
Auflösung, mit Traffic Manager 162–173
Authentifizierungsfehler 332
Auto Swap 46
Automation Hybrid Worker 272
AWS (Amazon Web Services) 191
az cosmosdb, Befehl anzeigen 152
az group, Befehl erstellen 131
az keyvault secret Befehl anzeigen 221
az keyvault, Befehl erstellen 212
az vm disk attach, Befehl 51
az vm resize, Befehl 127, 129
az vm, Befehl anzeigen 105
az vm, Befehl erstellen 51, 95, 197
az vm, Befehl 95
az-Speicherkonto, Befehl erstellen 210
Azure AD (Azure Active Directory) 222
Azure Application Gateway 108
Azure Application Insights 180
  Erstellen von Konten in 271–272
    Azure Automation-Pullserver und 280–282
    Definition 280–282
  PowerShell DSC 278–282
  Ressourcen 272–274
    Anzeigen der Ausgabe von 276–277
    Ausführung 276–277
    Beispiel von 274–277

Runbooks 272–274
Übersicht über 179, 269–274
Azure Automation 243, 269–283
Backup-Pläne 196–198
RPO (Recovery Point Objective) 194–195
RTO (Recovery Time Objective) 195–196
Richtlinien und Aufbewahrung 193–196
Komplette VM-Wiederherstellung 199–201
Wiederherstellung auf Dateiebene 199
Wiederherstellen von VMs 198–201
Azure Backup 191–201, 243
Azure Bastion 24, 115
Azure CLI 7, 12–13, 28, 31, 81–82, 152, 161
Azure Cloud Shell 12–13
Azure Cognitive Services 259–260
Azure Container-Instanz. Siehe ACI
Azure Container-Register. Siehe ACR
Azure DNS (Domain Name Service) 158–162
Azure DNS-Datensatztypen 160
Azure Event Grid 320–321
Azure Event Hubs 321–322
Azure Firewall 238
Azure Front Door 163–164
Azure Function Apps 318
Erstellen von Function Apps zur Analyse von Gerätedaten 328–331
Hub, zentrale Geräteverwaltung mit 303–309
Integration in Service Bus 322–325
Komponentenüberprüfung 315
Streaming von Hub-Daten zu Web-Apps 309–315
Übersicht über 300–302
Azure IoT (Internet of Things) 300–316
Azure IoT Edge 304–305
Erstellen von Schlüsseltresoren und Geheimnissen 219–221
Softwaretresore und HSMs 217–218
Daten in der Cloud sichern 216–221
Einspeisen von Zertifikaten 229–232
Erstellen von Zertifikaten 229–232
MSIs (Managed-Service-Identitäten) 221–229
Speichern von Schlüsseln 211–213
Übersicht von 211
Azure Key Vault 216–233, 304
Azure Kubernetes Service. Siehe AKS
Azure Logic Apps 318
Azure Machine Learning-Service 258
Azure Monitor 243
Anzeigen von geltenden NSG-Regeln 184–186
Erfassen von Netzwerkpaketen 186–188
IP-Datenflussüberprüfung 183–184
Azure Network Watcher 182–188
Erstellen von Netzwerkverbindungen für den Remote-Zugriff 68–69
Erstellen von VMs 69–70
Verwenden von SSH-Agents für die Verbindung zu VMs 70–72
Entwickeln von Beispielwebanwendungen mit sicherem Datenverkehr 68–72
Erstellen von NSG-Filterregeln 67–68
Erstellen von NSGs 64–65
Zuweisung von NSGs zu Subnetzen 66–67
Sichern und Steuern des Datenverkehrs mit NSGs 64–68
DNS-Auflösung 62–64
Erstellen virtueller Netzwerke 59
Öffentliche IP-Adressen 62–64
Unternetze erstellen 59
Virtuelle Netzwerkschnittstellenkarten 61
Virtuelle Netzwerkkomponenten 58–64
Azure Networking 58–72
Azure Portal 12
Azure PowerShell 11, 13, 31, 81–82, 161
Entwurf rund um den Anwendungslebenszyklus 76–77
Schutz von Ressourcen durch Sperrung 79–80
Schützen und Kontrollieren von Ressourcen 78–79
Verwalten und Gruppieren von Ressourcen mit Tags 80–81
Ansatz für 75–81
Erstellen 82–84
Erstellen von Mehrlingen von Ressourcentypen 84–85
Speichern 87
Tools für 85–86
Vorlagen für 81–87
Azure Resource Manager 75–89
Azure Security Center 234, 249
Azure Service Bus 310, 321–322
Azure Service Fabric 289
Azure Site Recovery 201–204, 243
Hinzufügen von Datenträgern zu VMs 50–52
Datenträger 49–50
Datenträger-Caching-Optionen 50
Standardspeicher im Vergleich zum Premium-Speicher 48–49
Temporäre Datenträger 49–50
VM-Speicher 47–50
Redundanz 56–57
Speicherverfügbarkeit 56–57
Tabellenspeicher 53–54
Warteschlangenspeicher 55–56
Vorteile von 52–57
Azure Storage 47–57
Azure Traffic Manager. Siehe Traffic Manager
OMS (Operations Management Suite) 243
Überprüfen und Anwenden von Updates 245–249
Azure Update Management 241–249
Azure-zu-Azure-Replikation 203
Bereitstellen von Anwendungen für Web-Apps, die mit mehreren Instanzen betrieben werden 140
Bereitstellungs-Slots und 35, 44–46
Bots erstellen mit LUIS 264–267
Bots mit LUIS ausführen 264–267
Diagnose-Protokollen, Anzeigen von 42–44
Erstellen von Netzwerkverbindungen für den Remote-Zugriff 68–69
Erstellen von VMs 69–70
Verwenden von SSH-Agents für die Verbindung zu VMs 70–72
Entwickeln mit sicherem Datenverkehr 68–72

Bereitstellen von HTML-Beispiel-Sites 39–42
Erstellen grundlegender Web Apps 37
Erstellen 37–42
Erstellen von Bots 260
Skalierung 127–139
Streaming von Azure IoT Hub-Daten in 309–315
Übersicht über 34–35
Unterstützte Sprachen und Umgebungen 34–35
Verwalten 42–44
Azure Web Apps 33–45
Problembehandlung 31–32
Speicher auf 18
Übersicht über 8–13
Azure Cloud Shell 12–13
Azure Portal 12
Azure PowerShell 13
Lokale Azure CLI 13
Verwaltungstools 11–13
Virtualisierung in 10–11
Azure-Plattform
Azure-Schnellstart-Vorlagen 7

B

Backend-IP-Pool, bei Load Balancern 107
Back-End-Pools 107, 116–119
Backslash-Zeichen 40, 68
Backup-Pläne 196–198
Backup-Pläne 196–198
Richtlinien und Aufbewahrung 193–196
Wiederherstellen von VMs 198–201
Azure Backup 191–201
Azure Site Recovery 201–204
Backups 191–204
Bash Shell 12
Bastion Host 23–24
Befehl „az vm list-sizes” 127
Befehle, Umbruch langer Zeilen 40
Benutzerdefinierte SSL-Zertifikate 207
Benutzungsdefinierte Bing-Suche 259
Bereitstellen von HTML-Sites 39–42
Bereitstellungs-Slots 44–46
Besitzer, Rolle 78
Betriebssystem des Rechenzentrums (DC/OS) 293
RPO (Recovery Point Objective) 194–195
RTO (Recovery Time Objective) 195–196
Bindung 193–196
Bing Autosuggest-Dienst 259
Blob Storage 52
Ausführung mit LUIS 264–267
Entwickeln mit LUIS 264–267
Erstellen 260
Bots für Web Apps
Branches, in Git 41
B-Serie, VM 18
Building the Web of Things (Guinard and Trifa) 315

C

Caret-Symbol 27
CD (Kontinuierliche Bereitstellung) 75
CI (Kontinuierliche Integration) 75
CLI (Befehlszeilenschnittstelle) 12
Erstellen von Schlüsseltresoren und Geheimnissen 219–221
Softwaretresore und HSMs 217–218
Clouds, Daten sichern in 216–221
Cluster mit AKS 294 – 295 Kollektionen 146
Compute-optimierte VM-Größen 17
Computer Vision Service 259
Concat-Funktion, Ressourcen-Manager 85
connectionString-Variable 307
ACI (Azure Container-Instanz) 289–292
Ausführen von Websites auf Kubernetes 295–297
Erstellung von Clustern mit 294–295
AKS (Azure Kubernetes Service) 293–297
Übersicht über 284–288
Container 146, 284–299
Container Orchestrator 293
Content Moderator Service 259
Copy-Funktion, Ressourcen-Manager 84
copyIndex()-Funktion 84, 98, 102, 104
Bereitstellen von Web Apps mit 156–157
Datenbanken erstellen und füllen 145–149
Erstellen von Konten und Datenbanken 145–152
Globale Redundanz hinzufügen 149–152
Skalieren von Datenbanken 143–144
Strukturierte Datenbanken (SQL) 142
Unstrukturierte Datenbanken (NoSQL) 142–143
Übersicht über 141–144
Zugriff auf global verteilte Daten 152–156
Cosmos DB 141–157
curl-Anfrage 226–227
Custom Script Extension 179

D

Data Science Virtual Machines (DSVMs) 258
database_password-Variable 228
Dateispeicher 53
Daten ruhen 208
Erstellen 145, 149–152
Füllen145–149
Globale Redundanz hinzufügen 149–152
bei Cosmos DB
Skalierung 143–144
Datenbanken
Datenbankserver, vertikale Skalierung für 126
Cachingoptionen 50
Datenträger 49–50
Hinzufügen zu VMs 50–52
temporäre 49–50
Datenträger
Datenträger 49–50
Definieren der Verteilung mit Load-Balancer-Regeln 112–114

Globale Verteilung auf die nächsten Instanzen 167–173
Erstellen von NSG-Filterregeln 67–68
Erstellen von NSGs 64–65
Zuweisung von NSGs zu Subnetzen 66–67
Sichern und Steuern mit NSGs 64–68
Weiterleitung von direktem Datenverkehr mit NAT-Regeln 114–116
Datenverkehr (Traffic)
Datenwissenschaftler, Tools für 257–259
DC/OS (Betriebssystem des Rechenzentrums) 293
DDoS (Distributed-Denial-of-Service) 182
default-allow-ssh-Regel 241
DenyAllInBound-Regel 67, 184
DenyAll-Regeln 185
deny-Zustand 238
dependsOn 104
Deploy-to-Azure, Schaltfläche 98
Diagnose-Log-Dateien 42–44
Direkter Datenverkehr, Routing 114–116
Distributed Denial of Service (DDoS) 182
DKIM (DomainKeys Identified Mail) 160
DNS-Auflösung 62–64, 158
Docker 284, 287
Docker Swarm 293
Dockerfiles 291–292
DomainKeys Identified Mail (DKIM) 160
aktualisieren 97–98
echte, delegieren an Azure DNS 160–162
Fehler 96–97
Domänen
DR (Disaster Recovery) 201
Drittanbietertools 86
DSC (Gewünschte Statuskonfiguration) 179, 278, 282–283
DSCMs (Data Science Virtual Machines) 258
dynamische Zuweisung 62

E

Echte Domänen delegieren 160–162
ein hohes SLA-Niveau 19, 183
Einspeisen von Zertifikaten 229–232
enableHttpsTrafficOnly-Parameter 210
Endpoint Discovery 153
Endpunkte 323
Endpunkt-Überwachungsprotokoll 168
Entscheidungsdienst 259
Erfassen von Netzwerkpaketen 186–188
Erweiterungen 305
ETW (Event Tracing for Windows) 180
Event-Endpunkt 310, 312

F

Face Service 259
Federal Information Processing Standard (FIPS) 218
Fehlerdomänen 96–97
Fehlermeldungen 31
Filtern 67–68
FIPS (Federal Information Processing Standard) 218
Forum, für dieses Buch 5
FQDN (vollständig qualifizierter Domänenname) 63
Frei/Gemeinsamer Service-Plan 36
Front-End-IP-Pools 107–110
Füllen von Datenbanken 145–149
Function Apps 328–331
Funktionsschlüssel 332
Gardner, Lyza Danger 315
Erhalten aus VMs mit MSI 224–229
Erstellen 219–221

G

Geheimnisse
Geografisches Routing 163–164
Georedundanter Speicher (GRS) 56
Geschützte VMs, löschen 205
Gewichtete Routing-Methode, Traffic Manager 163
Bereitstellen von HTML-Beispielsites 39–42
Kennworts für, Zurücksetzen des 314
Lern- 37
Git 12
git push azure master, Befehl 156
git push dev master, Befehl 45
Azure Automation und Quellcodesteuerung mit 274
Azure-Schnellstartvorlagen auf 87
Kontos für, Erstellen eines 7
Repository für dieses Buch 5
Ressourcen 333
Übersicht von 39
GitHub
Global verteilte Daten 152–156
Globale Redundanz 149–152
Erstellen von Traffic Manager-Profilen 164–166
Globale Verteilung des Datenverkehrs auf die nächsteInstanz 167–173
Globale Weiterleitung, mit Traffic Manager 162–173
Google Maps-Beispiel 256
GPU (Grafikprozessoreinheit) 267
GPU-VM-Größen 17
Größenänderung von VMs 126–127
GRS (Georedundanter Speicher) 56
Grundlegender Service-Plan 36
Gruppieren von Ressourcen 80–81
Guinard, Dominique D. 315

H

HashiCorp 86
Hochleistungs-SSDs 18
HPC (High Performance Computing) 267
HSM (Hardwaresicherheitsmodule) 212, 217–218
HTML-Sites, bereitstellen 39–42
HTTP 20, 168, 206
HTTP-pfadbasierter Modus, Integritätstests 110
HTTPS 20, 168, 206
Hyper-V 15

I

IaaS (Infrastructure-as-a-Service) 9, 14, 33–34
IaC (Infrastructure as Code) 82
IIS (Internet Information Services) 29, 233
Images, VM 16–17
IMDS (Instance Metadata Service) 222
Infrastructure-as-a-Service (IaaS) 9, 14, 33
Infrastructure-as-Code (IaC) 82
  Erstellen von Netzwerkressourcen über Verfügbarkeitszonen hinweg 94–95
  Erstellen von VMs in Verfügbarkeitszonen 95
Infrastrukturredundanz mit Verfügbarkeitszonen 95
Inkrementelle Sicherung 193
install, Befehl 27
Installation von Webservern 24–27
Instance Metadata Service (IMDS) 222
Instanzen, erstellen 290–292
  Erstellen 110–112
  Konfigurierung 110–112
  Übersicht von 107
Integritätstests
Interaktiver Zugriff auf die Boot-Konsole 176
Interner Load Balancer 108
Internet Information Services (IIS) 29, 233
Internet Load Balancer 108
Intervallparameter, Integritätstests 111
iotconnectionstring-Variable 312
IP-Adressbereiche 60
IP-Datenflusses, Überprüfen des 183–184
IP-Datenflussüberprüfung 183–184
IP-Pools 107
IPv4-Adressen 109
IPv4-Host-Einträge 160
IPv6-Adressen 109
IPv6-Host-Einträge 160
Isolierte Umgebungen 36

J

JavaScript on Things (Gardner) 315
JIT-Aktualisierungen (Just-in-time) 237–249
JQ-Parser 226
JSON (JavaScript Object Notation) 82–83, 86
JWT (JSON Web Token) 226
  Azure Cognitive Services 259–260
  LUIS 261–264
  Machine Learning und 254–259
  Übersicht über 254–255
    Ausführung mit LUIS 264–267
    Entwickeln mit LUIS 264–267
    Erstellen 260
  Web-App-Bots

K

KI (Künstliche Intelligenz) 253–268
  Integritätstests 110–112
  VMs mit Load Balancern 119–122
Konfigurierung
    Erstellen 145–149, 152
    Füllen 145–149
    Globale Redundanz hinzufügen 149–152
  bei Cosmos DB
  in Azure Automation, Erstellen 271–272
Konten
Kontingente 102, 132
Kontinuierliche Bereitstellung (CD) 75
Kontinuierliche Integration (CI) 75
    Erstellen von NSG-Filterregeln 67–68
    Erstellen von NSGs 64–65
    Zuweisung von NSGs zu Subnetzen 66–67
  Datenverkehr mit NSGs 64–68
  Ressourcen 78–79
Kontrolle
  Siehe auch AKS
Kubernetes 293, 295–299
Kubernetes in der Praxis (Luksa) 298

L

LAMP-Webserver 27, 72
Lassen Sie uns das Projekt verschlüsseln 207
LCM (Local Configuration Manager) 278
Learn Docker in a Month of Lunches (Nutzen Sie Ihre Mittagspause, um mehr über Docker zu erfahren) (Stoneman) 297
Learn Git in a Month of Lunches (Git in der Mittagspause kennenlernen) (Umali) 37
Lebenszyklen von Apps 76–77
Leistungsbedingungen, Warnmeldungen zu 181–182
Leistungsmetriken 178–182, 188
Lernmaterialien 333
Lese-/Schreib-Caching 50
Lese-Cacherichtlinie 50
Leser, Rolle 78
  Ausführen von Web Apps unter 34
  Verwenden von DSC mit 282–283
Linux
  Definieren der Datenverkehrsverteilung mit Regeln 112–114
  Erstellen und Konfigurieren von VMs mit 119–122
  in der Praxis 120–122
    Definieren der Datenverkehrsverteilung mit Load Balancer-Regeln 112–114
    Erstellen von Front-End-IP-Pools 108–110
    Integritätstests 110–112
    Weiterleitung von direktem Datenverkehr mit NAT-Regeln 114–116
    Zuweisen von Gruppen von VMs zu Backend-Pools 116–119
  Komponenten von 106–119
Load Balancer 94
Local Configuration Manager (LCM) 278
Log Analytics-Arbeitsbereiche 243
Logic Apps 35, 182, 325–328
Lokal redundanter Speicher (LRS) 56
Long Term Support (LTS) 22

Löschen geschützter VMs 205
LRS (Lokal redundanter Speicher) 56
LTS (Long Term Support) 22
  Ausführen von Web-App-Bots mit 264–267
  Erstellen von Web-App-Bots mit 264–267
  Übersicht über 257–264
LUIS (Language Understanding Intelligent Service)
Luksa, Marko 298

M

Managed-Object-Format-Datei (MOF) 280
Marketplace, Azure 7
Maschine Learning. Siehe ML
Maven 12
Message Analyzer von Microsoft 186
Message Analyzer, Microsoft 186
  Azure Event Grid 320–321
  Azure Event Hubs 321–322
  Azure Service Bus 321–322
  Entwicklung von Service Bus 322–325
  Integration von Service Bus mit IoT Hubs 322–325
Messaging-Plattformen 319–325
Metrikwarnungen 182
Mitwirkender, Rolle 78
Mitwirkungsrolle bei virtuellen Maschinen 79
Mitwirkungsrolle für die Website 79
  Azure Cognitive Services 259–260
  Beziehung zur künstlichen Intelligenz 257–259
  Künstliche Intelligenz und 254, 256
  LUIS (Language Understanding Intelligent Service) 261–264
  Tools für Datenwissenschaftler 257–259
  Übersicht über 255–256
    Ausführung mit LUIS 264–267
    Entwickeln mit LUIS 264–267
    Erstellen 260
  Web-App-Bots
ML (Machine Learning) 253–268
Module 270
MOF-Datei (Managed Object Format) 280
Monolithische Anwendung 288
MSIs (Managed-Service-Identitäten) 221–229

N

Nachrichtentext, Eigenschaft 56
Namensserver-Datensätze 160
NAT (Netzwerkadressübersetzung) 107, 114–116
Network Watcher 184
Networking. Siehe Azure Networking
Netzwerkpakete 186–188
Netzwerkressourcen 94–95
Netzwerkschnittstellenkarten (NICs) 61
Netzwerksicherheitsgruppen. Siehe NSGs
Netzwerkverbindungen für den Remote-Zugriff 68–69
  Routing 158–174
  Verwalten 158–174
Netzwerkverkehr
NICs (Netzwerkschnittstellenkarten) 61, 117
–no-self-perms-Parameter 220
NoSQL (Unstrukturierte Datenbanken) 142–143
Notfallwiederherstellung (DR) 201
  Anzeigen von geltenden Regeln 184–186
  Erstellen 64–65, 118
  Erstellen von Filterregeln 67–68
  im Azure Security Center 234
  Sichern und Steuern des Datenverkehrs mit 64–68
  Übersicht von 112
  Zuweisung zu Subnetzen 66–67
NSGs (Netzwerksicherheitsgruppen) 20
Nummerierung, auf 0 basierend 99
Nur Anwenden-Modus, DSC 279
nx-Modul 283

O

Öffentliche IP-Adressen 20, 62–64, 94, 108
OMS (Operations Management Suite) 243, 272

P

PaaS (Platform-as-a-Service) 10, 33, 37, 137
parallele VMs 102
Parameter „-A” 121
Parameter 82, 84, 89
Perl, Programmiersprache 34
Personalisierter Service 259
PHP 34
Platform-as-a-Service (PaaS) 10, 33
Pointer-Aufzeichnungen 160
  Back-End 116–119
  Front-End-IP-Pools 108–110
Pools
portbasierter Modus, Integritätstests 110
  Azure Automation-Pullserver und 179, 280–282
  Definition 280–282
PowerShell DSC (Desired State Configuration) 278–282
PowerShell. Siehe Azure PowerShell
Premium-Service-Plan 36
Premium-SSD-Festplatten (Solid State Drive) 18–19
Priorisierte Routing-Methode, Traffic Manager 163
Private IP-Adressen 108
Privater Schlüssel, eines SSH-Schlüsselpaars 20–22, 71
Privater Schlüssel, eines SSH-Schlüsselpaars 71
    Anzeigen von geltenden NSG-Regeln 184–186
    Erfassen von Netzwerkpaketen 186–188
    IP-Datenflussüberprüfung 183–184
  Azure Network Watcher 182–188
  Azure-Plattform 31–32
  Leistungsmetriken 178–182
  VM-Diagnose 175–177
  Warnungen 178–182
Problembehandlung 175
Produktionsslot 46
Profile, in Traffic Manager 164–166

Protokolle. Siehe Diagnose-Protokolle
Pullserver 280–282
Python, Programmiersprache 28, 34

Q

Quellcode 5

R

RA-GRS (Schreibgeschützter georedundanter Speicher) 57
Raspberry Pi 306–309
RBAC (Rollenbasierte Zugriffssteuerungen) 78, 161, 184, 211
RDP-Verbindung (Remotedesktop-Protokoll) 20, 71
readLocations 153
  Übersicht über 56–57
  von VMs mit Verfügbarkeitsgruppen 96–102
  Vorteile von 90–91
Redundanz
Redundanz. Siehe auch Infrastrukturredundanz, mit Verfügbarkeitszonen
Regeln für die automatische Skalierung 133–136
Remote-Desktop-Protokoll-Verbindung (RDP) 20, 71
remotes 41
Repräsentativer Staatstransfer (REST) 31
  Bereinigen 30
  Horizontal skalieren 128–129
  Kontrollieren 78–79
    Gruppieren 80–81
    Verwalten 80–81
  mit Tags
  Schutz durch Sperren 79–80
  Schützen 78–79
Ressourcen 5, 7
Ressourcengruppen 315
Ressourcentypen 84–85
REST (Repräsentativer Staatstransfer) 31
REST-APIs 161
  RPO (Recovery Point Objective) 194–195
  RTO (Recovery Time Objective) 195–196
Richtlinien 193–196
Rollenbasierte Zugriffssteuerung (RBAC) 78, 161, 211
Rollentrennung 62
Routing nach Leistung 163–164
RPO (Recovery Point Objective) 193–195
RTO (Recovery Time Objective) 193, 195–196
Run-As-Konten 272
  Anzeigen der Ausgabe von 276–277
  Ausführen 182
  Ausführung 276–277
  Übersicht über 272–274
Runbooks, für Azure Automation 274–277

S

SaaS (Software-as-a-Service) 10
SAS-Token (Shared Access Signature) 87
  Erstellen von Schlüsseltresoren 219–221
  Speichern der Schlüssel in Azure Key Vault 211–213
Schlüssel
Schlüsselpaare 20
Schnittstellenkarten 61
Schreibgeschützter georedundanter Speicher (RA-GRS) 57
    Erstellen von NSG-Filterregeln 67–68
    Erstellen von NSGs (Netzwerksicherheitsgruppen) 64–65
    Zuweisung von NSGs zu Subnetzen 66–67
  Datenverkehr mit NSGs 64–68
  Ressourcen 78–79
Schutz
Schutz von Ressourcen 79–80
Schwellenwertparameter, Integritätstests 111
Security Center Overview-Fenster 236
Sender Protection Framework (SPF) 160
Serielle VMs 102
Server Message Block (SMB) 53
  Erstellen von Function Apps zur Analyse von IoT-Gerätedaten 328–331
  Erstellen von Logic Apps 325–328
  GitHub-Ressourcen 333
    Azure Event Grid 320–321
    Azure Event Hubs 321–322
    Azure Service Bus 321–322
    Entwicklung von Service Bus 322–325
    Integration von Service Bus mit IoT Hubs 322–325
    Übersicht über 317–319
  Messaging-Plattformen 319–325
  Messaging-Plattformen 319–325
Serverloses Computing 317–333
  Erstellen 322–325
  Integration in IoT Hubs 322–325
Service Bus
Service Level Agreements (SLAs) 247
Servicedaten 160
Service-Endpunkte 146
Service-Pläne für Apps 35–38
servicePrincipalName 224
Serviceprinzipal 222
Session-Affinitätsmodus 112–113
Shared-Access-Signature-Token (SAS) 87
  Erstellen von Netzwerkverbindungen für den Remote-Zugriff 68–69
  Erstellen von VMs 69–70
  Verwenden von SSH-Agents für die Verbindung zu VMs 70–72
sicherem Datenverkehr, Entwickeln von Web-Apps mit 68–72
Sicherheit 115
Sinks 180
  Skalieren von Web-Apps 136–139
    Erstellen 131–133

Erstellen von Regeln für die automatische Skalierung 133–136
VM-Skalierungsgruppen 129–136
Horizontales Skalieren von Ressourcen 128–129
Vertikales Skalieren von VMs 125–127
Vertikales Skalieren von Web-Apps 127–128
Vorteile von 124–129
Skalierbare Apps 124–140
Datenbanken 143–144
Ressourcen horizontal 128–129
Größenänderung von VMs 126–127
Herunterskalieren 127
VMs vertikal 125–127
von VMs 127
Übersicht über 136–139
Vertikal 127–128
Web Apps
Skalierung
Erstellen 131–133
Erstellen von Regeln für die automatische Skalierung 133–136
Skalierungsgruppen, für VMs 129–136
SLAs (Service Level Agreements) 247
SMB (Server Message Block) 53
Software-as-a-Service (SaaS) 10
Softwaretresore 217–218
SONiC (Software for Open Networking in the Cloud) 11
Speicher (vDisk) 11
Speichern von Vorlagen 87
Speicheroptimierte VM-Größen 17
Speicheroptimierte VM-Größen 17
Sperren 79–80
SPF (Sender Protection Framework) 160
Sprachdienst 259
Sprachdienst 259
Sprechererkennungsdienst 259
SQL (Structured Query Language) 53, 142
SSE (Storage Service Encryption) 209–210
Agents für die Verbindung zu VMs 70–72
Verbindung zu VMs mit 24–27
SSH (Secure Socket Shell)
ssh-keygen, Befehl 21
SSH-Schlüsselpaare 20–22
SSL-Zertifikat 207
Staging 41
Standard-HDDs 18
Standardkontingente 102
Standard-Service-Plan 36
Standard-SSDs 18–19
Standorte für den Endpunkt 153
Startdiagnose 175–177
Start-of-Authority-Datensätze (SOA) 160
Statische Zuordnung 63
auf Azure 18
Datenträger 49–50
Datenträger-Caching-Optionen 50
Standardspeicher im Vergleich zum Premium-Speicher 48–49
Temporäre Datenträger 49–50
in VMs 47–50
Redundanz 56–57
Verfügbarkeit von 56–57
Warteschlangenspeicher 55–56
Storage
Streaming von IoT-Hub-Daten 309–315
Streaming von Protokolldateien 43
Structured Query Language (SQL) 53, 142
Strukturierte Daten 144
Strukturierte SQL-Datenbanken 142
Erstellen 59
Zuweisung von NSGs zu 66–67
Subnetze
Subnetzebene 185
Suchdienst 259
Gruppieren von Ressourcen mit 80–81
Verwalten von Ressourcen mit 80–81

T

Table Storage 52 Tags
Temporäre Datenträger 49–50
Terraform 86
Test in Web-Chat-Option 266
Testintervall für Bereitstellungpunkte 168
Time to Live (TTL) 167
Bereich der Problembehandlung 183
Bereitstellen von Web Apps für 174
Erstellen von Profilen in 164–166
Globale Verteilung des Datenverkehrs auf die nächsten Instanzen 167–173
Globales Routing und Auflösung mit 162–173
Traffic Manager
Tresore, Schlüssel- 219–221
Trifa, Vlad M. 315
TTL (Time to Live) 167
Überprüfung von Updates 245–249
Übersetzer-Text-Service 259
Anzeigen von geltenden NSG-Regeln 184–186
Erfassen von Netzwerkpaketen 186–188
IP-Datenflussüberprüfung 183–184
Azure Network Watcher 182–188
Leistungsmetriken 178–182
VM-Diagnose 175–177
Warnungen 178–182

U

Überwachung 175
Ubuntu Linux 14, 26
Umali, Rick 37
Universal Coordinated Time (UTC) 196
Universelle VM-Größen 17
Unstrukturierte Daten 144
Unterstützte Sprachen 34–35
Update Management Overview-Fenster 243
Update-Domänen 96
UTC (Universal Coordinated Time) 196

V

Variablen 82, 84, 89, 271
Verbindungen 270
  Anzeigen der Verteilung von VMs 101–102
  Verteilen von VMs über 98–101
    Fehlerdomänen 96–97
    Update-Domänen 97–98
  VM-Redundanz mit 96–102
Verfügbarkeitsgruppen 91
  Erstellen von Netzwerkressourcen über 94–95
  Erstellen von VMs in 95
  Infrastrukturredundanz mit 95
Verfügbarkeitszonen 91
  ruhend 208–209
  Schlüssel in Azure Key Vault 211–213
  SSE (Storage Service Encryption) 209–210
  Übersicht über 206–208
  von VMs 211–214
Verschlüsselung 206–215
Verwaltete Datenträger 18
VHD (Virtuelle Festplatte) 53
Virtualisierung 10–11
Virtuelle CPU (vCPU) 11
Virtuelle Festplatte (VHD) 53
virtuelle Maschinen. Siehe VMs
  DNS-Auflösung 62–64
  Erstellen 59
  Öffentliche IP-Adressen 62–64
  Schnittstellenkarten 61
  Unternetze erstellen 59
Virtuelle Netzwerke 58–64
Virtuelle Netzwerkkonnektivität (vNIC) 11
Virtuelle NIC-Ebene 185
Virtuelle private Netzwerke (VPNs) 19, 36, 38
Virtueller Arbeitsspeicher (vRAM) 11
Vision-Dienst 259
Visual Studio-Editor 85–86
  Anzeigen der Verteilung in Verfügbarkeitsgruppen 101–102
  Aufheben der Zuweisung 30
  Bereitstellen aus Vorlagen 102–105
  Diagnose 175–177
  Diagnoseerweiterungen 178
  Erhalten von Geheimnissen aus MSI 224–229
    aus Webbrowsern 22
    Bereinigen von Ressourcen 30
    in Verfügbarkeitszonen 95
    Kosteneinsparungen und 18
    mit Load Balancern 119–122
    Problembehandlung auf Azure 31–32
    Windows-VM 29–30
  Erstellen 14–32, 69–70
  Größen von 17
  Größenänderung 126–127
  Herunterskalieren 127
  Hinzufügen von Datenträgern zu 50–52
  Installation von Webservern 24–27
    Azure Storage 18–19
    Virtuelle Vernetzung 19–20
    VM-Größen 17–18
    VM-Images und 16–17
  Konfiguration 15–20
  Konfigurieren mit Load Balancern 119–122
  Löschen 30
    Fehlerdomänen 96–97
    Update-Domänen 97–98
  Redundanz mit Verfügbarkeitsgruppen 96–102
    Erstellen 131–133
    Erstellen von Regeln für die automatische Skalierung 133–136
    Installieren von Anwendungen auf 139
  Skalierungsgruppen 129–136
  SSH-Schlüsselpaar, Erstellen für die Authentifizierung 20–22
    Datenträger 49–50
    Datenträger-Caching-Optionen 50
    Standardspeicher im Vergleich zum Premium-Speicher 48–49
    Temporäre Datenträger 49–50
  Storage 47–50
  Über Verfügbarkeitsgruppen verteilen 98–101
    mit SSH 24–27
    mit SSH-Agents 70–72
  Verbinden mit 120–122
    Lab 214–215
    Speichern der Schlüssel in Azure Key Vault 211–213
  Verschlüsselung von 211–214
  Vertikales Skalieren 125–127
    Komplette VM-Wiederherstellung 199–201
    Wiederherstellung auf Dateiebene 199
  Wiederherstellen 198–201
    Erstellen von Regeln, um Webverkehr zuzulassen 28
    Webserver in der Praxis anzeigen 28–29
  Zulassen, dass Webverkehr möglich ist 27–29
  Zuweisen von Gruppen von zu Back-End-Pools 116–119
VMs (Virtual Machines)
VM-Skalierungsgruppe mit einer VM 130
VMware 15
Vom Benutzer zugewiesene verwaltete Identitäten 222
Vom System zugewiesene verwaltete Identitäten 222
  Erstellen 82–84
  Erstellen von Mehrlingen von Ressourcentypen 84–85
  Speichern 87
  Tools für 85–86
Vorlagen, für Azure Resource Manager 81–87
VPNs (Virtuelle private Netzwerke) 19, 36, 38, 183

W

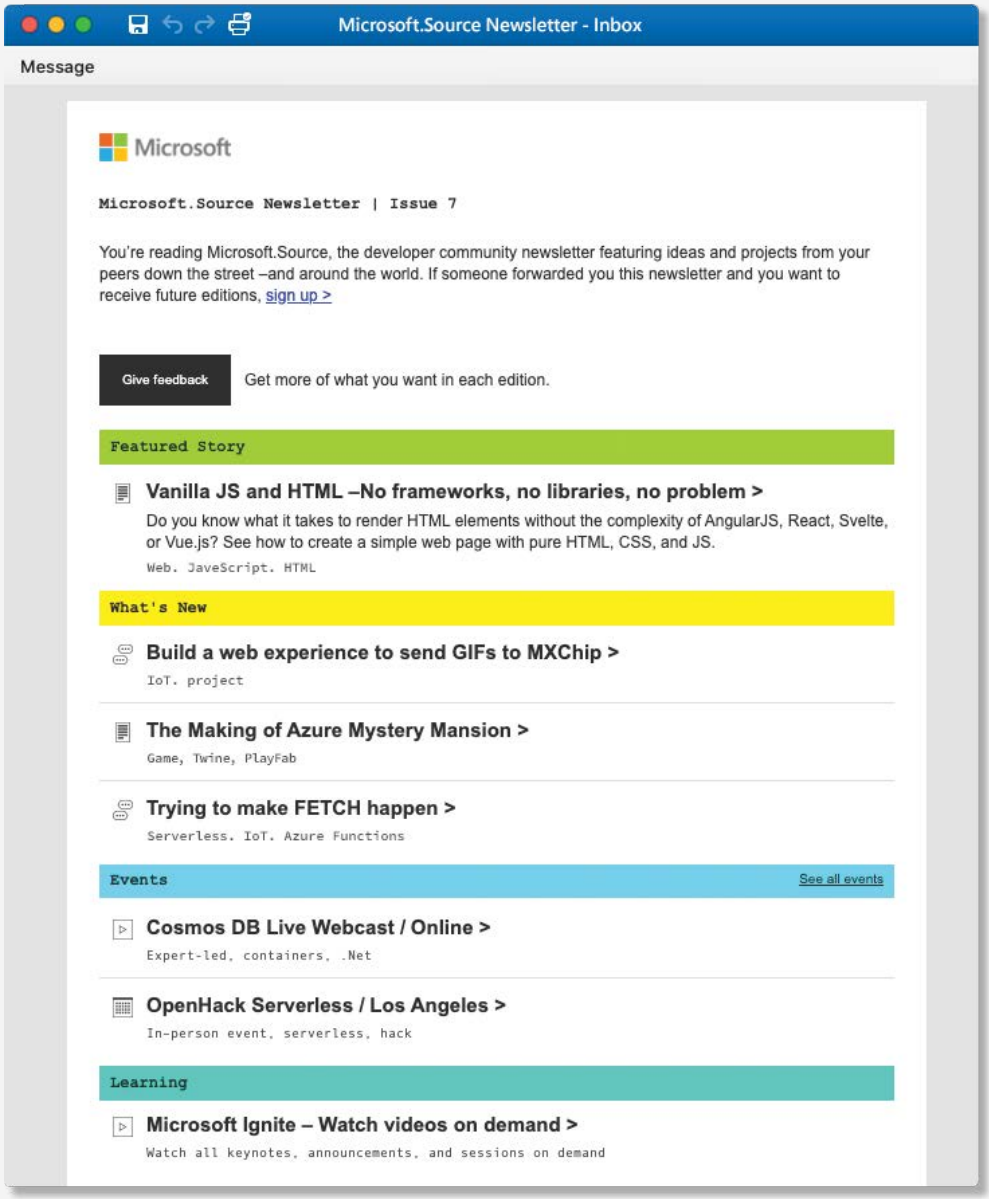
Warnungen 178–182
Warteschlangenspeicher 53, 55–56
  Azure Storage 18
  VM-Größen 17
Webbrowser, Erstellen von VMs aus 22
Webhooks 274
  in der Praxis 28–29
  Installieren 24–27

Webserver
Websites, auf Kubernetes ausgeführte 295–297
WebSockets 312
  Erreichen von VMs zulassen 27–29
  Regeln erstellen zum Zulassen 28
Web-Traffic
Wechsel mit Vorschau 46
Weiterleitung von direktem Datenverkehr mit NAT-Regeln 114–116
  Komplette VM-Wiederherstellung 199–201
  Wiederherstellung auf Dateiebene 199
Wiederherstellen von virtuellen Maschinen 198–201
Wiederherstellung auf Dateiebene 199
Wiederherstellungspunkte 193
Wiederherstellungszeitziel (RTO) 193
Windows, Ausführen von Web-Apps unter 34

Z

Zeitpläne 134, 270
  Einspeisen 229–232
  Erstellen 229–232
Zertifikate 270
Zonendienste 93
–Zonenparameter 95
Zonenredundante Dienste 93
ZRS (Zonenredundanter Speicher) 56

Microsoft



By developers, for developers

Microsoft.Source newsletter

Get technical articles, sample code, and information on upcoming events in Microsoft.Source, the curated monthly developer community newsletter.

- Keep up on the latest technologies
- Connect with your peers at community events
- Learn with hands-on resources

Sign up