



Azure verstehen

EIN LEITFADEN FÜR ENTWICKLER



Zusammenfassung

Jetzt ist die beste Zeit, Entwickler zu sein. Mit der Cloud können Sie Lösungen erstellen, die nie zuvor möglich gewesen sind. Innerhalb von Minuten sind Sie dazu in der Lage, neue Features für Millionen von Benutzern freizugeben. Sie können die Grenzen der gegenwärtigen technologischen Beschränkungen binnen Tagen überwinden. Entwickler wiederum brauchen nur wenige Monate, um Ideen von Grund auf erfolgreich für das Geschäft umzusetzen. Unternehmen benötigen Anwendungen, die es ihnen gestatten, ihre Kundenbindung zu optimieren und sich von Mitbewerbern zu differenzieren. Teams müssen mit Agilität bei der Anwendungsentwicklung und einer kürzeren Time-to-Market punkten. Daher benötigen Entwickler eine flexible Plattform, die sowohl ein Scale-up und Scale-down basierend auf geschäftlichen Anforderungen ermöglicht als auch solide Ressourcen liefert, die Ausfällen standhalten können.

Microsoft Azure bietet Entwicklern eine Cloud-Plattform für die Entwicklung innovativster Anwendungen.

Dieser Leitfaden beschreibt das „Warum“ und „Wie“ von Szenarien, die sich für die Cloud eignen. Der Fokus liegt auf der Erstellung von Anwendungen, die die in Microsoft Azure verfügbaren Plattform-Services verwenden. Die zweite Hälfte dieses Leitfadens zeigt die Breite und Tiefe der Azure-Plattform und -Services, die dafür geschaffen worden sind, Entwickler bei der Erstellung wirklich wettbewerbsfähiger und differenzierter Anwendungen zu unterstützen.

Zur Zielgruppe für diesen Leitfaden gehören:

- Entwickler, die verstehen möchten, warum Azure die beste Cloud-Plattform zur Erstellung von Anwendungen ist und wie sie schnell damit beginnen können – auf Basis der heute zu erstellenden Anwendungen.
- Technische Führungskräfte, die Azure in Erwägung ziehen, um die Entwicklung neuer oder bestehender Anwendungen zu unterstützen.

VERÖFFENTLICHT VON:

Microsoft Corporation
One Microsoft Way
Redmond, Washington 98052-6399, USA

Copyright © 2016, Microsoft Corporation

Alle Rechte vorbehalten.

Die Informationen in diesem Dokument werden im vorliegenden Zustand zur Verfügung gestellt. Die Informationen und Ansichten in diesem Dokument, einschließlich URLs und anderer Referenzen zu Internet-Webseiten, können sich ohne Vorankündigung ändern. Sie tragen das Risiko der Verwendung.

Einige Beispiele dienen lediglich der Illustration und sind frei erfunden. Reale Bezüge sind weder beabsichtigt noch erwünscht.

Dieses Dokument gibt Ihnen keine Rechte auf geistiges Eigentum in Microsoft-Produkten. Sie dürfen dieses Dokument zu internen Zwecken und als Referenz kopieren und verwenden.



Inhaltsverzeichnis

Einleitung.....	4
Die sich verändernde Welt der Anwendungsentwicklung	4
Azure: die Cloud-Plattform, die für Dich – den Entwickler – konzipiert wurde	5
App Service: Web- und mobile Anwendungen.....	6
Service Fabric: Microservice-basierte Anwendungen.....	7
Cloud Services: Monolithische Anwendungen.....	7
Azure Functions: Ereignisgesteuerte Ausführung.....	7
Zusammenfassung	8
Szenario 1: Erstellen einer Webanwendung.....	9
A. Eine einfache Webanwendung erstellen	9
B. Skalierung einer Webanwendung.....	12
C. Isolieren einer Webanwendung.....	14
Szenario 2: Backend für eine mobile Anwendung erstellen	16
A. Ihrer Webanwendung mobile Features hinzuzufügen.....	16
B. Benutzerverhalten analysieren mit Ihrem mobilen Backend.....	18
Szenario 3: Erstellung eine Internet of Things (IoT) -Anwendung	20
Szenario 4: Erstellen einer maßgeschneiderten Microservice-basierten Anwendung.....	23
Azure App Service	25
Azure Service Fabric.....	26
Azure Cloud Services	27
Azure Functions.....	28
Docker und Container	30
Azure Container Service.....	30
Datenbank-Services	32
SQL-Datenbank.....	32
DocumentDB	32
HDInsight HBase.....	32
Azure-Tabellen	32
Andere Daten-Services.....	33
Erweiterte Analysen & IoT.....	33
Cortana Intelligence Suite.....	33
Internet of Things (IoT): Internet der Dinge.....	34
Entwickler-Services.....	35
Visual Studio Team Services.....	35
Dev-Test Labs.....	35
HockeyApp.....	35
Application Insights.....	35
Identität.....	36
Unterstützung von Azure AD als eine Möglichkeit zur Anmeldung bei Ihrer Anwendung	36
Suche nach Benutzern, Verwaltung der Benutzerprovisionierung und Kontrolle des Zugriffs auf Ihre Anwendung.....	36
Erweiterte Sicherheitsfunktionen	36
Microsoft Graph.....	36
Integration	37
BizTalk Services.....	37
Hybride Verbindungen.....	37
Service Bus.....	37
Queues.....	37
Media Services & CDN.....	38
Media Services	38
Gängige Szenarien.....	38
Empfohlene nächste Schritte.....	39

Überblick

Einleitung

Die Cloud verändert Erwartungen. Ihre Kunden erwarten mehr. Ihr Unternehmen erwartet mehr. Erwarten Sie mehr.

Bekannte Unternehmen wie Uber und Facebook wurden in der Cloud geboren. In nahezu jeder Branche treibt die Einführung der Cloud das Wachstum voran. Diese Unternehmen bewegen sich schneller, liefern mehr Wertschöpfung und erfüllen unsere wechselnden Bedürfnisse effektiver als es mit der Einschränkung traditioneller Ansätze möglich ist. Sie finden neue Wege, um die Flexibilität der Cloud zu ihrem Vorteil zu nutzen. Sie entwickeln Möglichkeiten, fast unendlich zu skalieren und tiefere Einblicke in ihre Kunden zu erhalten - etwas, von denen andere Unternehmen nur träumen können.

Nun möchten Sie in die Cloud wechseln. Und Sie fragen sich, wie das geht. Sie wollen die Vorteile, wissen aber nicht, wie Sie beginnen und Fallstricke vermeiden können, mit denen die Cloud-Pioniere anfänglich konfrontiert waren?

Der [britische Computer-Wissenschaftlicher David Wheeler](#) sagte einst: „Alle Probleme in der Informatik können durch eine weitere Dereferenzierungsebene gelöst werden“. Und in der Tat, sobald Unternehmen herausfanden, wie bei gleichzeitiger Kostenkontrolle eine massive Skalierung erfolgen konnte, war es nur eine Frage der Zeit, bis dieser Schicht zu Cloud Computing kommen würde.

Diese Ebene ist die Anwendungsplattform – und diese ist bereits heute vorhanden.

Die Wartung, Skalierung und Redundanz Ihrer Anwendung können Sie weiterhin über eine Farm von Standardservern steuern, die sich an Ihre Bedürfnisse angepasst ausdehnt und zusammenzieht. Aber jetzt

können Sie die Bereitstellung auf einer Plattform vornehmen, die Ihnen all das abnimmt. Dies ist die „zweite Welle“ des Cloud Computing. Dort konzentrieren Sie sich darauf, die Innovation zu liefern, die Ihre Kunden fordern – und nicht auf die Infrastruktur, die erforderlich ist, um diese auszuführen.

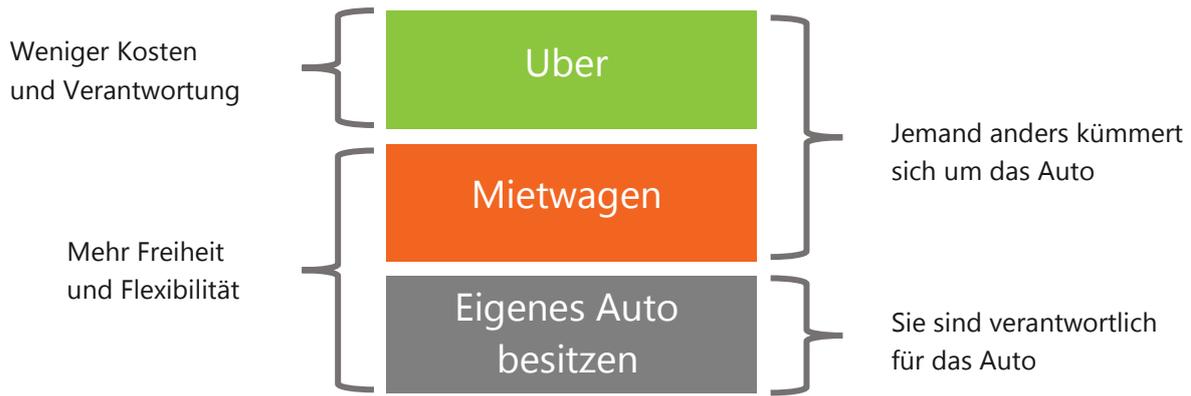
Dieser Leitfaden ist eine Einführung in die Azure-Anwendungsplattform. Er bietet Ihnen die Anleitungen und Richtungen, die Sie benötigen, um mit der Erstellung neuer Anwendungen zu beginnen oder ihre vorhandenen Anwendungen in die Cloud zu verschieben.

Die sich verändernde Welt der Anwendungsentwicklung

Am Anfang gab es physische Maschinen. Sie waren teuer und Entwickler luden mehrere Anwendungen auf ihnen, um Kosten zu sparen. Dies beendete keine Konflikte und verursachte unerwartete Fehler, während die effektive Auslastung der Maschinen nach wie vor niedrig war. Entwickler kämpften damit sicherzustellen, dass Änderungen an einer Anwendung nicht dazu führten, dass eine andere, damit nicht zusammenhängende Anwendung nicht mehr funktionierte.

Dann kamen virtuelle Maschinen. Da mehrere virtuelle Maschinen auf einer einzigen physischen Maschine ausgeführt werden können, installierten Entwickler ihre Anwendungen in geschlossenen virtuellen Maschinen. Zwar reduzierte das Konflikte und verbesserte die Auslastung, doch führte dies auch zu Shared-Memory und unkalkulierbarer Leistung. Und da die Provisionierung selbst für die größten Belastungen gemacht werden musste, würde ein „Scaling-in“ nicht wirklich Geld sparen.

Überblick



Dann kam die Cloud. Plötzlich konnten Unternehmen virtuelle Maschinen mieten, wenn sie sie brauchten, und ein Scale-up oder Scale-down von Anwendungen ganz nach Bedarf vornehmen. Noch immer allerdings mussten Patches und Updates auf Betriebssystemebene installiert und konfiguriert werden, und es gab Schwierigkeiten mit dem Low-Level-Networking. Doch Unternehmen führten Infrastruktur-as-a-Service (IaaS) ein und machten sich dessen Flexibilität erfolgreich zunutze, um die Geschwindigkeit und Skalierbarkeit zu verbessern, bei gleichzeitiger Kostenkontrolle. Die meisten Vorteile dieses Modells entstanden allerdings beim Betrieb und der Infrastruktur, nicht bei der Anwendungsentwicklung. Da immer mehr Unternehmen erkennen, dass Anwendungsinnovationen aufgrund der Kunden- und wettbewerbsorientierten Nachfrage schneller als je zuvor erfolgen müssen, verlagert sich der Fokus weg von herkömmlichen Infrastrukturkosteneinsparungen und stattdessen darauf, wie die Anwendungsentwicklung produktiver erfolgen kann.

Azure: die Cloud-Plattform, die für Dich – den Entwickler – konzipiert wurde

Die Geschichte von Microsoft ist eine Geschichte von Entwicklern. Kein anderer Cloud-Anbieter kann auf eine solche reiche Tradition bei der Entwicklung und Entwicklern wie Microsoft zurückblicken. Das allererste Microsoft-Produkt im Jahr 1975 war ein BASIC-Interpreter, und seitdem haben wir uns unermüdlich auf die Bedürfnisse von Entwicklern konzentriert. Wir erstellen nicht nur die weltweit besten Plattformen und Entwickler-Tools, wir benutzen sie auch selber, um damit Software und Services für unsere Kunden zu erstellen. Und wir erstellen nicht nur die weltweit Entwickler-freundlichste Cloud, sondern wir

benutzen sie auch selber. Wir sind ein „Cloud-first“-Unternehmen, das den Fokus auf die Erstellung von Cloud-Produkten und -Services der nächsten Generation legt. Unsere Services müssen jederzeit verfügbar sein, für Millionen von Benutzern skalieren und sich nahtlos mit neuen Funktionen aktualisieren lassen. Genau wie unsere Kunden benötigen wir eine Anwendungsplattform, die für uns „hart arbeitet“. Azure haben wir konzipiert, um die Entwicklerproduktivität zu steigern.

Was also unterscheidet eine Entwicklerplattform von einer Infrastrukturplattform?

Es ist ziemlich einfach, denn es gibt Ebenen der Verantwortung mit Vorteilen und Kompromissen zwischen ihnen. Nehmen wir zum Beispiel ein Auto als Analogie: Wenn Sie ein Auto besitzen, müssen Sie den Kaufprozess durchlaufen, eine Versicherung abschließen, es über einen langen Zeitraum hinweg unterhalten und warten, sowie für eine Park- oder Unterstellmöglichkeit sorgen.

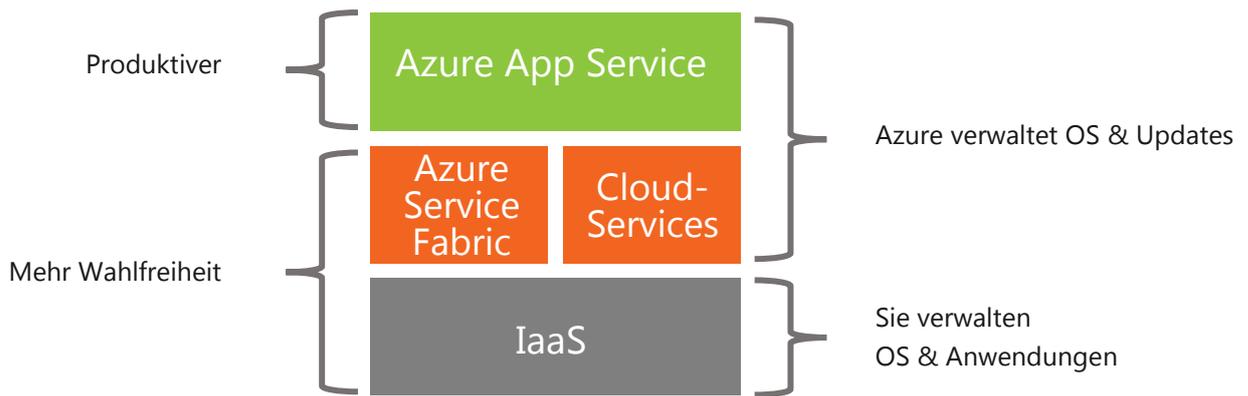
Mieten Sie ein Auto, erhalten Sie den Vorteil eines Autos für einen festen Zeitraum ohne den „Eigentums-Overhead“. Und natürlich ist ein Uber oder ein Taxi die

Alaska®

Alaska Airlines benötigte eine interne Webanwendung, um für mobile Mitarbeiter verfügbare Standby-Reisen von außerhalb der Unternehmens-Firewall zu buchen. Die Entwickler nahmen die bestehende Logik und verschoben die Anwendung nach Azure App Service, kreierte mit Xamarin ein mobiles Frontend und stellten Mitarbeitern die Anwendung in Rekordzeit zur Verfügung.

[Mehr zu Alaska und Azure](#)

Überblick



wahrscheinlich preisgünstigste Option, allerdings mit Grenzen bei der Flexibilität – möglicherweise müssen Sie fünf Minuten Wartezeit in Kauf nehmen oder draußen im Regen stehen, während Sie auf die Fahrt warten.

Sie können mehr aus Ihrer knappen Entwicklungszeit herausholen durch die Verwendung einer Anwendungsplattform, bei der die Komplexität sowie die Verantwortung für die Wartung herausgenommen worden sind. Möglicherweise gibt es einige Kompromisse bei der Flexibilität, aber insgesamt können Sie schneller mehr erledigen mit einer niedrigeren Total-Cost-of-Ownership, aufbauend auf Platform-as-a-Service (PaaS).

(Es gibt zwar noch einige Situationen, in denen es sinnvoll ist, [Anwendungen auf IaaS zu erstellen](#), doch das werden wir später besprechen.)

Kommen wir nun zu den Details mit einer kurzen Tour durch die wichtigsten [Azure-Plattform-Services](#). Wir zeigen Ihnen, wo und wann Sie diese verwenden können. Dann führen wir Sie durch einige verbreitete Kundenanwendungs-Szenarien, was Ihnen dabei hilft, die besten Designentscheidungen zu treffen.

App Service: Web- und mobile Anwendungen



Azure App Service umfasst eine Reihe von Services, die alles haben, was Sie zum Erstellen von Anwendungen sowohl für Web- als auch mobile Clients über ein einziges Anwendungs-Backend benötigen. Mit .NET, Node.js, Java, PHP

und Python werden eine Vielzahl an Sprachoptionen unterstützt. Azure App Service besteht aus *Web-Apps*, *mobile Apps*, *Logik-Apps* und *API-Apps*. Web-Apps und mobile Apps bieten Backend-Services für Web- und mobile Anwendungen. API-Apps erlauben es, APIs verfügbar zu machen, die von beliebigen Anwendungen sicher genutzt werden können und ihre Anwendung mit Dutzenden populärer Dienste wie Office 365 und

Salesforce.com verbinden. Logik-Apps schließlich gestatten Ihnen die Automatisierung von Geschäftsprozessen sowie die Workflow-Koordinierung mithilfe von Tools, die von keiner Code-Erfahrung bis zur kompletten Leistung von Visual Studio reichen. Ein vollständig integriertes DevOps ermöglicht Ihnen die Bereitstellung von Anwendungsaktualisierungen mittels eingebauter Funktionen zum Staging, Rollback, Tests in der Produktionsumgebung sowie Leistungstests. Alle Aspekte Ihrer Anwendungen können Sie in Echtzeit und historisch mit detaillierten Betriebsprotokollen überwachen. App Service beschäftigt sich mit der darunter liegenden Infrastruktur und beseitigt für Sie die Notwendigkeit, Ihre Infrastruktur zu warten und zu patchen.

Wofür Sie es benutzen können

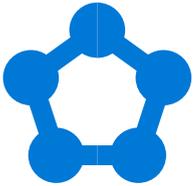
App Service ist die bevorzugte Option zur Erstellung von Web- oder mobilen Backends, da es Produktivitäts-, Skalierungs-, Leistungs- und Bereitstellungsoptionen für die meisten Anforderungen bietet. Benutzen Sie App Service, wenn Sie sich darauf konzentrieren möchten, einzigartige Funktionen für Ihre Anwendungen zu erstellen und dafür einfach eine Infrastruktur benötigen, die arbeitet. Zusätzlich ist es hilfreich für die Erstellung einer API-Ebene, die für Kunden, Partner und Mitarbeiter

Jet.com benötigte eine extrem schnelle Innovation zur Erstellung einer effektiven E-Commerce-Plattform. Mit einer leeren Leinwand beginnend, konnten sie eine beliebige Plattform auswählen. Dank der reichhaltigen Anwendungs-Plattform-Services von Microsoft Azure war das Unternehmen in der Lage, das Release binnen 12 Monaten anstatt von zwei Jahren vorzunehmen.

[Mehr zu Jet.com und Azure](#)

Überblick

unter Verwendung des zugehörigen API-Management-Services sicher verfügbar gemacht werden kann. App Service können Sie zusammen mit etlichen anderen Azure-Services (zum Beispiel Redis Cache oder Azure Traffic Manager) nutzen, um leistungsfähige Anwendungen schnell zu erstellen und bereitzustellen.



Service Fabric: Microservice-basierte Anwendungen

Azure Service Fabric ist eine Plattform, die Entwickler und ISVs nutzen können, um kundenspezifische Microservice-

basierte Anwendungen im Cloud-Maßstab sowie mit 24x7-Verfügbarkeit zu erstellen und zu verwalten. Microservices stellen einen Architekturansatz auf Basis feingradiger, lose gekoppelter Services dar – was fortgeschrittene Entwickler-Agilität, Cloud-Skalierung, Selbstheilungs-Widerstandsfähigkeit sowie Always-on-Verfügbarkeit ermöglicht. Service Fabric bildet die Grundlage für viele Microsoft-Services wie Azure SQL-Datenbank, Azure DocumentDB, Cortana, Windows Intune und Skype for Business. Wir stellen Ihnen exakt dieselbe Technologie zur Verfügung. Nutzen Sie Service Fabric überall dort, wo Sie Ihre Microservice-basierten Anwendungen ausführen möchten – ob in Azure, Ihrem eigenen Rechenzentrum oder in anderen Clouds, was Ihnen maximale Flexibilität verleiht.

Wofür Sie es benutzen können

Das primäre Service Fabric-Szenario ist die Erstellung von in hohem Maße skalierbaren Always-on-Services, die sich (wie die zuvor erwähnten Microsoft-eigenen Services) ohne Ausfallzeiten aktualisieren lassen. Service Fabric verwenden Sie, wenn Sie eine neue Anwendung erstellen oder eine existierende Anwendung umschreiben. Dadurch machen Sie sich Microservices zunutze, haben sich zur „Cloud-first“-Architektur bekannt und können die erforderlichen laufenden Versionsaktualisierungen ohne Ausfallzeiten durchführen. Sie profitieren von verteilter Skalierbarkeit und hoher Leistung ebenso wie von Lese- und Schreibzugriffen auf Daten mit geringer Latenz.

Cloud Services: Monolithische Anwendungen



Azure Cloud Services war das ursprüngliche, in 2010 gelaunchte Platform-as-a-Service (PaaS) -Angebot von Microsoft. Inspiriert von den

Bedürfnissen von Entwicklern, ermöglicht Cloud Services Ihnen die leichte Bereitstellung von .NET-Anwendungen. Dabei können Sie sich die Vorteile der Cloud-Infrastruktur wie virtuelle Maschinen und virtuelle Netzwerke zunutze machen, ohne Low-level-Details verwalten oder sich mit der Konfiguration von Windows Server sowie der darunter liegenden Infrastruktur beschäftigen zu müssen.

Wofür Sie es benutzen können

Cloud Services ist eine effiziente Lösung, wenn Sie direkten Zugriff auf die darunter liegende Infrastruktur benötigen oder Internet Information Services (IIS) direkt konfigurieren müssen. Wie bei App Service brauchen Sie sich hier mit dem Patching oder der Aktualisierung des zugrunde liegenden Betriebssystems nicht zu beschäftigen. Manchmal werden die Cloud Services auch zur Migration bestehender Webanwendungen mit Abhängigkeiten zu IIS-Konfigurationen oder erforderlichen Drittanbieter-Komponenten genutzt – da sie eine automatische Skalierung bieten und dennoch zahlreiche Entwickler-orientierte Plattformvorteile erlauben. Seit Einführung der Cloud Services haben sich die Design-Optionen erheblich weiterentwickelt. App Service und Service Fabric spiegeln einen moderneren PaaS-Ansatz wider, den Sie als die Grundlage für Ihre Anwendungsentwicklung betrachten sollten.

Azure Functions: Ereignisgesteuerte Ausführung



Azure Functions bieten Serverless-Compute für ereignisgesteuerte Lösungen, die die vorhandene Azure-Anwendungsplattform um die Möglichkeit zur Implementierung von Code erweitern, der von Ereignissen getriggert wird. Diese können in anderen Azure-Services, SaaS-Produkten (wie Office365 und Salesforce.com) sowie on-premises Systemen auftreten. Bei Azure Functions bezahlen Sie nur dann, wenn Ihre Funktion tatsächlich ausgeführt wird. Es bietet eine intuitive, Browser-basierte Benutzeroberfläche, die es Ihnen gestattet, geplante oder ausgelöste Code-Teile zu erstellen, die in einer Vielzahl von Programmiersprachen implementiert sind.

Wofür Sie es benutzen können

Wenn Sie unabhängige, unbeaufsichtigte Funktionen zur Reaktion auf Ereignisse erstellen möchten, die über Azure, SaaS-Produkte und Ihre on-premises Services hinweg



Container

Container haben für Entwickler oberste Priorität, da sie überall eine agile und effiziente Ausführung von Anwendungen ermöglichen. Microsoft arbeitet eng mit Partnern wie Docker und Mesosphere zusammen, um Lösungen in diesem Bereich anzubieten. Container werden oft auf IaaS-Ebene verwendet, sind aber ebenfalls beliebt für die Implementierung Microservices-basierter Lösungen wie Service Fabric. In diesem Leitfaden lesen Sie mehr zu [Docker und Containern](#) und dem [Azure Container Service](#).

auftreten. Azure Functions können Sie verwenden, um Ihren Anwendungen organisatorische oder Funktionen hinzuzufügen, die für Mehrwert sorgen, ohne dass diese direkt eine Modifikation Ihrer Code-Basis erfordern. Erlaubt zum Beispiel Ihre familienfreundliche Website die Veröffentlichung eigener Fotos, könnten Sie eine Azure-Funktion hinzufügen, die kontinuierlich den Speicherort für neue Bilder abfragt und einige der fortgeschrittenen Azure-APIs verwendet, um Bilder mit unangemessenen Inhalten zu identifizieren und diese Fotos mit einer geschwärzten Version zu ersetzen. All dies lässt sich ohne jegliche Code-Änderungen der bestehenden Anwendung realisieren und komplett eigenständig ohne Auswirkungen auf die vorhandene Anwendung bereitstellen.

Zusammenfassung

Als Entwickler sind Sie immer neugierig auf neue Technologien. Sie möchten experimentieren mit neuen und interessanten Ideen, um Ihre Anwendungen besser zu machen oder neue Anwendungen zu erstellen. Bisher mussten Sie die Hauptarbeit häufig selbst erledigen - Ihre eigenen Funktionen schreiben oder ein Feature oder einen Service für jede Anwendung, die Sie geschrieben haben, neu erstellen.

Doch jetzt können Sie moderne Lösungen mit wenigen Zeilen Code implementieren. Neben einer reichhaltigen Anwendungsplattform verfügt Azure über leistungsstarke Dienste wie Machine-Learning, mobile Einbindung und Analysen. Auf dem Weg dorthin bauen Sie Benachrichtigungen in Ihre Anwendungen direkt mit ein. Ihren eigenen mobilen Benachrichtigungsdienst können Sie natürlich jederzeit selbst schreiben, aber warum sollten Sie nicht von vorgefertigten Plattform-Services

profitieren und Ihre Zeit stattdessen Funktionen widmen, die Ihre Anwendung besser machen?

Bei Azure finden Sie Open-Source- und Cross-Plattform-Unterstützung über die breiteste Auswahl an Programmiersprachen, Frameworks, Tools, Datenbanken und Architekturen hinweg. Das ermöglicht es Ihnen, die maximale Reichweite für jede Zeile Code zu erzielen. "Einmal schreiben, überall ausführen" ist schon immer ein Branchen-Mantra gewesen, doch mit Azure verfügen Sie über unvergleichliche Möglichkeiten durch Produkte wie Azure Stack. Dieser erlaubt Ihnen die Code-Ausführung an weiteren Orten, ohne Änderung. Nutzen Sie die Tools, die Sie lieben, sowie die Fähigkeiten, die Sie bereits haben – und führen Sie praktisch jede Anwendung, unter Verwendung Ihrer gewünschten Datenquelle, auf Ihrem jeweiligen Betriebssystem und auf Ihrem bevorzugten Gerät aus.

Der moderne Entwickler ist ein *Lösungs-Komponist*, der kundenspezifischen Code mit den besten fertigen Services zur Erstellung von Anwendungen kombiniert, die in dieser Form vor Kurzem nicht vorstellbar waren. Die Bereitstellung dieser Anwendungen erfolgt mit erstaunlicher Schnelligkeit. Unternehmen auf der ganzen Welt nutzen Azure zur Erstellung neuer Anwendungen und Erweiterung bestehender Anwendungen, um ihr Geschäft zu unterstützen und zu transformieren. Wie können Sie der nächste sein?

TalkTalk

TalkTalk TV, der drittgrößte Kabel-TV-Anbieter Großbritanniens, entschied sich dafür, Teile seiner IaaS-Anwendung mit Azure Service Fabric neu zu gestalten, um die Vorteile einer Microservices Architektur zu nutzen. Die neue Content-Management- und Auflösungs-Plattform erlaubte es ihnen, schnellere Lieferzyklen zu erzielen und die Verfügbarkeit bei Aktualisierungen aufrechtzuerhalten.

[Mehr zu TalkTalk und Azure](#)

Gängige Szenarien für Cloud-Anwendungen

Sie sind bereit für den Wechsel in die Cloud. Sie möchten die Vorteile nutzen, aber sind unsicher, wie und wo Sie beginnen sollten. Die folgenden Szenarien sind gedacht zur Orientierung sowie als Ausgangspunkt für gängige Anwendungs-Workloads, die viele Kunden haben.

Hinweis: Achten Sie darauf, die neueste Version des Azure SDK herunterzuladen. Prüfen Sie [diesen Link](#), um alle verfügbaren Downloads zu finden.

Szenario 1: Erstellen einer Webanwendung

Die Erstellung einer Webanwendung ist das vielleicht gebräuchlichste Anwendungsmuster. In diesem Abschnitt sehen wir uns an, wie sich eine bestehende Webanwendung migrieren, skalieren und schützen lässt.

A. Eine einfache Webanwendung erstellen

Übersicht

Sie haben eine interne Branchenanwendung mit einem Web-Frontend und einem relationalen Datenbank-Backend. Die Anwendung muss zwar nicht für Millionen von Benutzern skalieren, stellt für Ihr Geschäft aber eine kritische Komponente dar. Derzeit ist die Webanwendung nur innerhalb der Firewall zugänglich, aber Sie würden es begrüßen, wenn Personen die Anwendung auch von außerhalb des LANs erreichen könnten – unter Beibehaltung strenger Sicherheitsprinzipien.

Das Problem

Eine Anwendung selbst zu hosten, ist mit ganz realen, teilweise versteckten, Kosten verbunden. Einige dieser Kosten enthalten möglicherweise manuelle bereitgestellte Aktualisierungen, das Patchen von Betriebssystemen und die Verwendung von außerhalb der Firewall erforderlichen VPN-Verbindungen. Die Kosten sind nicht überwältigend und verbergen sich wahrscheinlich in den Gesamt-Budgets sowie aktuell zugewiesenen Arbeitslasten. Doch das macht die Kosten nicht weniger real.

Die Lösung

Für diese Art der Anwendung ist es relativ einfach, sie nach Azure zu verschieben, und die Vorteile können umfangreich sein. Bei einem on-premises Szenario müssen Entwickler möglicherweise Infrastruktur-Teams arbeiten, wann immer neue Versionen der Anwendung auszurollen sind – zur Validierung, dass die Anwendung keine anderen, potenziell kritischeren Anwendungen stört. In Azure hingegen können Entwickler Aktualisierungen mittels Continuous Delivery früher und häufiger bereitstellen mit on-premises und Online-Versionskontrollsystemen wie TFS, GitHub, BitBucket und Visual Studio Team Services.

Verwendete Services

- *Azure Web-Apps*
- *Azure SQL-Datenbank*
- *Azure Active Directory*

Gängige Szenarien für Cloud-Anwendungen

Das Patchen von Betriebssystem ist eine notwendige Aufgabe. Eine schlecht verwaltete Infrastruktur kann schwerwiegende Sicherheitskonsequenzen oder Auswirkungen auf die Software haben. Bei Azure werden die Maschinen, die sich unterhalb der Azure Web-Apps befinden, für Sie automatisch gepatcht und verwaltet.

Der Zugriff auf Unternehmens-Websites, die sich außerhalb einer Firewall befinden und eine Authentifizierung benötigen, erfordert üblicherweise irgendeine VPN-Verbindungsform. Bei Azure hingegen ermöglicht es die Azure Active Directory-Integration in Ihr on-premises Active Directory den jeweiligen Benutzern, sich mit denselben Anmeldeinformationen zu verbinden und dabei sogar die Multifaktor-Authentifizierung zu verwenden – gänzlich ohne die Notwendigkeit für eine VPN-Verbindung.

Das „Wie“

Instinktiv überlegen Sie möglicherweise, Ihre virtuellen Maschinen nach Azure zu verschieben, SQL Server einzurichten und IIS zu konfigurieren. Es kann Sie mehr Zeit kosten, bis alles genau so konfiguriert ist, wie Sie es wünschen, doch Sie haben die Kontrolle, dies zu tun. Ihr Team wird für das Patchen, die Wartung sowie die Administration zuständig sein müssen. Mit diesem Lift-and-Shift-Modell erhalten Sie dieselbe Funktionalität, die Sie von lokalen Bereitstellungen erwarten würden. Es kann sogar der schnellste Weg in die Cloud sein. Doch die Kosten- und Ressourcenanforderungen sind wahrscheinlich ähnlich wie zu Ihrer on-premises Lösung, nur dass diese jetzt in der Cloud gehostet würde.

Stattdessen können Sie virtuelle Maschinen überspringen und eine [Azure-Webanwendung in App Service](#) verwenden, um dort Ihre Webanwendung sowie [Azure SQL-Datenbank](#) für Ihre Datenbank zu hosten und Ihre Webanwendung mit [Azure Active Directory](#) zu schützen. Dies bedeutet, dass Sie die Vorteile von Continuous Delivery-Optionen nutzen können, um Ihre Webanwendung schneller in Betrieb zu nehmen und im Laufe der Zeit zu warten.

Eine Webanwendung in der Cloud hosten

Unabhängig davon, ob Ihre Anwendung in .NET, Java, PHP, Node.js oder Python geschrieben ist, sind Sie dazu in der Lage, Ihre Webanwendung für eine Azure Web App in App Service bereitzustellen. Sie können mehrere Web-, mobile, API- und Logik-Apps in App Service bereitstellen. Für die meisten Webanwendungsbedürfnisse ist App Service die Antwort.

- Ihre Webanwendung in einer Azure Web App [bereitstellen](#), [kopieren](#) oder [migrieren](#).
- [Laden Sie ein SSL-Zertifikat hoch](#) und [konfigurieren Sie einen benutzerdefinierten Domänennamen](#) für die Webanwendung.
- [Continuous Delivery im Webanwendungs-Repository](#) über BitBucket, TFS, GitHub oder Mercurial.

Sie können Azure Active Directory für die Synchronisierung mit dem on-premises Active Directory Ihres Unternehmens einrichten und Benutzer mit der Webanwendung über die Windows-Authentifizierung von außerhalb der Firewall verbinden.

- [Azure AD Connect](#) herunterladen und installieren zur Integration von on-premises Identitäten.
- Funktionen von Azure AD Connect konfigurieren wie die [Filterung](#) oder [Kennwortsynchronisierung](#).
- [Aktivierung der Verzechnissynchronisierung](#) für Azure Active Directory im Portal.
- [Authentifizierung aktivieren](#) mit Azure Active Directory für die Webanwendung und Azure SQL-Datenbank im Azure Management Portal.

Speicherung der Daten in der Cloud

Azure SQL-Datenbank ist eine verwaltete relationale SQL-Database-as-a-service (DBaaS) -Lösung in Azure – [und unterscheidet sich von der Ausführung von SQL Server in einer Azure Virtual Machine](#). Sie brauchen keine virtuellen Maschinen, Betriebssysteme und Datenbank-Software zu verwalten oder sich Gedanken um Aktualisierungen, Hochverfügbarkeit und Sicherungen zu machen. Im Allgemeinen kann Azure SQL-Datenbank die Anzahl der von einer einzelnen IT- oder Entwicklungsressource verwalteten Datenbanken sehr stark erhöhen.

Gängige Szenarien für Cloud-Anwendungen

Profi-Tipp

Implementieren Sie eine gestaffelte Bereitstellung, um die Staging- und Produktions-Slots Ihrer Webanwendung gegen keine Ausfallzeiten auszutauschen. A/B-Tests führen Sie in der Produktionsumgebung mit einem konfigurierbaren Bruchteil Ihres Live-Datenverkehrsaufkommens aus.

- [Bereitstellung in Azure SQL-Datenbank mit SQL Server Management Studio](#), [Exportieren der Datenbank in eine BACPAC-Datei, dann importieren oder BCP verwenden](#), oder verwenden Sie die [SQL Server-Transaktionsreplikation](#) mit minimaler Ausfallzeit.
- Sobald die Datenbank in Azure ist, können Sie sich damit über SQL Server Management Studio auf dieselbe Art und Weise verbinden, wie Sie es auch on-premises machen.
- Kopieren Sie die Verbindungszeichenfolge und platzieren Sie diesen im Verbindungszeichenfolgenbereich für die Webanwendung in Azure oder anderen Ressourcen, die sich damit verbinden.

Weiterführende Ressourcen

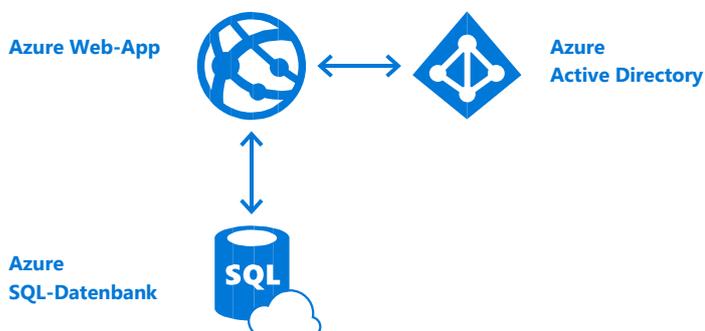
[Bereitstellen Ihrer Anwendung in Azure App Service](#) →

[Continuous Delivery für Azure mithilfe von Visual Studio Team Services](#) →

[Integrieren Ihrer lokalen Identitäten in Azure Active Directory](#) →

[Migrieren einer SQL Server-Datenbank zu Azure SQL-Datenbank in der Cloud](#) →

Architekturdiagramm



Gängige Szenarien für Cloud-Anwendungen

B. Skalierung einer Webanwendung

Die Erstellung einer simplen Webanwendung in Azure ist denkbar einfach. Jedes wachsende Unternehmen möchte jedoch dazu in der Lage sein, zu skalieren und eine globale Reichweite für seine Webanwendungen zu bieten.

Die Lösung

Zur Skalierung Ihrer Azure-Webanwendung in App Service brauchen Sie lediglich eine horizontale Leiste zu verschieben, um die Anzahl der Instanzen der verfügbaren Webanwendungen zu erhöhen. Natürlich möchten Sie auch ein konsistentes Erlebnis für alle Benutzer, gleich ob sich Ihr Kunde in einem Internet-Café in Rabat in Marokko oder in einem Hotel in Mexico City befindet. Dazu wählen Sie ein Azure-Rechenzentrum aus, in dem Ihre Webanwendung und Datenbank bereitgestellt werden. Azure Traffic Manager wird dann Benutzer zu der Instanz Ihrer Anwendung weiterleiten, die ihrem jeweiligen Standort am nächsten ist. Die geografische Expansion hat hohe Priorität für Azure, das im Laufe der Zeit um die Unterstützung neuer Regionen erweitert wird.

Das „Wie“

Eine Möglichkeit, um Skalierbarkeit und Verfügbarkeit zu erzielen, ist durch die Verwendung virtueller Maschinen, Load-Balancer und redundanter Ressourcen. Wenn Ihre Webanwendung allerdings immer verfügbar sein soll, müssen Sie jedoch zahlen, damit die virtuelle Maschine und Replikate immer verfügbar sind.

Auf die gleiche Weise, mit der App Service die Last der Verwaltung der Infrastruktur beseitigt, erleichtert es auch die Skalierung. Verwenden Sie eine Azure-Webanwendung zum Hosten Ihrer Webanwendung sowie die Datenbank in Azure SQL-Datenbank, und nutzen Sie Azure Traffic Manager, um Benutzer an Kopien der Webanwendung global weiterzuleiten. Aktivieren Sie die aktive Georeplikation in Azure SQL-Datenbank zur Synchronisierung von Datenbank-Replikas. Verwenden Sie Azure Redis Cache zur Verbesserung der Leistung, indem häufig benötigte Daten und Statusinformationen geladen werden.

Skalierung und Replizierung einer Anwendung

Azure Web-Apps ermöglicht Ihnen die Skalierung der maximalen Anzahl von Instanzen Ihrer Anwendung mit einem Schieberegler, wahlweise nach oben oder unten. Für dieses Scale-up oder Scale-Down legen Sie einen Zeitplan fest (etwa während Geschäftszeiten) oder geben einen CPU-Zielprozentsatz vor und Azure erledigt den Rest. Danach stellen Sie denselben Code für Webanwendungen rund um die Welt schnell bereit.

- Ihre Webanwendung als Azure Web App in App Service [bereitstellen](#), [kopieren](#) oder [migrieren](#).
- Geplante Zeiten festlegen für die [automatische Skalierung der Webanwendung nach oben oder unten \(Scale-up oder Scale-down\)](#) oder wählen Sie die Skalierung entsprechend der CPU-Last aus.
- Erstellen Sie mindestens eine andere Webanwendung in einer anderen Region.
- Gibt es Verweise zu einer anderen Datenbank, aktualisieren Sie die Verbindungszeichenfolgen.

Azure Web-Apps bietet bereits Failover- sowie Round-Robin-Datenverkehr-Routing-Funktionalität für Webanwendungen innerhalb eines Rechenzentrums. Traffic Manager gestattet es, ein Failover und Round-Robin-Datenverkehr-Routing für Webanwendungen in unterschiedlichen Rechenzentren festzulegen.

Globale Benutzer an eine Webanwendung weiterleiten

Azure Traffic Manager erlaubt es Ihnen, die Verteilung von Benutzerdatenverkehr an Ihre festgelegten Endpunkte – wie zum Beispiel Webanwendungen – zu steuern. Der Service verwendet bei Domain Name System (DNS) -Anfragen für die Domännennamen Ihrer Internet-Ressourcen eine intelligente Richtlinien-Engine, sodass Sie Ihre Webanwendungen überall in Rechenzentrum ausführen lassen können.

- Erstellen Sie ein eindeutiges Traffic Manager-Profil und [wählen Sie die Traffic Manager-Routingmethode aus](#).
- Auf der Endpunkte-Registerkarte fügen Sie die Webanwendung hinzu, auf die verwiesen werden soll.

Gängige Szenarien für Cloud-Anwendungen

- Legen Sie die Überwachungseinstellungen für das Traffic Manager-Profil fest, um sicherzustellen, dass die Endpunkte verfügbar sind. Sie können das Protokoll, den Port und den relativen Pfad angeben.
- Ändern Sie die Konfiguration so ab, sodass der [Domännennamen Ihres Unternehmens](#) auf einen Traffic Manager-Domännennamen zeigt.

Verwendete Services

- *Azure Web-Apps*
- *Azure SQL-Datenbank*
- *Azure Traffic Manager*
- *Azure Redis Cache*

Weltweite Speicherung

Azure SQL-Datenbank unterscheidet sich von SQL Server in einer Azure Virtual Machine, da es die aktive Georeplikation bietet. Hierbei erfolgt eine asynchrone Replikation von Transaktionen mit ausgeführtem Commit aus einer Datenbank auf bis zu vier Kopien der primären Datenbank auf verschiedene Server. Nutzen Sie dieses Feature zur Notfallwiederherstellung oder verwenden Sie eine online lesbare sekundäre Datenbank als Load-Balancer für Nur-Lese-Workloads zur Bedienung von Clients, die über verschiedene Regionen verteilt sind.

- [Bereitstellung Ihrer Datenbank in Azure SQL-Datenbank mit SQL Server Management Studio, die Datenbank in eine BACPAC-Datei exportieren, dann importieren oder BCP verwenden](#) oder [SQL Server-Transaktionsreplikation](#) mit minimaler Ausfallzeit verwenden.
- Skalierung auf den Premium Service-Tier, um den Standort auszuwählen, zu dem die SQL-Datenbank repliziert wird.
- In der Georeplikation [fügen Sie mindestens eine sekundäre Datenbank hinzu](#) ebenso wie die Zielregion und den Zielservers.

Profi-Tipp

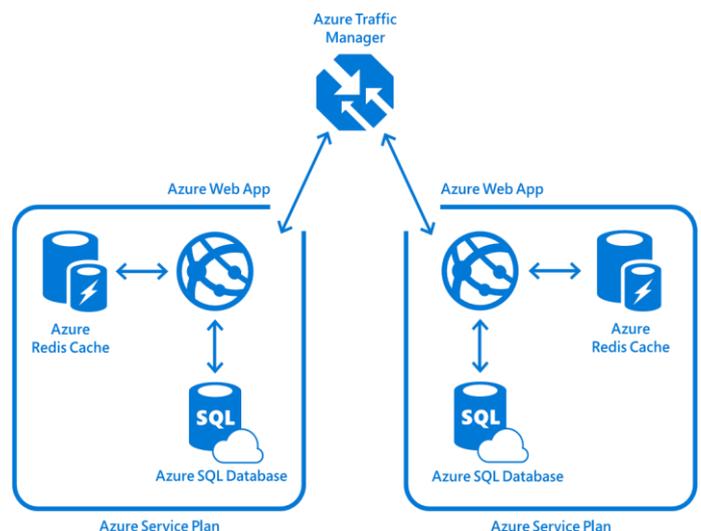
Erfahren Sie in dieser Azure Friday-Session, wie mit Azure Traffic Manager die virale Site „How-Old.NET“ erstellt und skaliert wurde.

Reaktionsfähigkeit der Anwendung steigern

Azure Redis Cache basiert auf dem beliebten Open-Source Redis-Cache, einem erweiterten Schlüsselwertspeicher, der Sie mit Werten (Zeichenfolgen, Hashes, Listen, Sets und sortierten Sets) arbeiten lässt. Er besitzt Master-Subordinate-Replikation sowie weitere leistungssteigernde Features. Diese separate verteilte Cache-Ebene erlaubt es, Ihren Daten-Tier für eine effizientere Nutzung von Compute-Ressourcen in Ihrer Anwendungsebene unabhängig zu skalieren.

- Erstellen Sie den Cache und [konfigurieren Sie den Cache-Client](#).
- [Verbindung mit dem Cache](#) und Speicherung der Anmeldeinformationen im Verbindungszeichenfolgenbereich der Webanwendung.
- [Objekte hinzufügen und abfragen](#) aus dem Cache sowie den Ablauf von Elementen im Cache spezifizieren.
- [Cache-Diagnosen aktivieren](#), um die Cache-Integrität zu überwachen.

Architekturdiagramm



Weiterführende Ressourcen

[Traffic Manager-Überwachung](#)



[Aktive Georeplikation für Azure SQL-Datenbank](#)



[Verwenden von Azure Redis Cache](#)



Gängige Szenarien für Cloud-Anwendungen



C. Isolieren einer Webanwendung

Viele Unternehmen sind mit der Erfordernis konfrontiert, Anwendungen zu schützen, die personenbezogene Daten (PII) enthalten. Eine einfache, öffentlich in Azure gehostete Webanwendung wird Ihren Sicherheitsanforderungen möglicherweise nicht gerecht. Stattdessen benötigen Sie eine abgeriegelte Umgebung mit einer Türe zur Außenwelt sowie eine aktive Abwehr möglicher Bedrohungen.

Die Lösung

Hostet Ihr Unternehmen sensitive Informationen, hat eine vollständig isolierte und dedizierte Umgebung ausschließlich für die Anwendungen Ihres Unternehmens oberste Priorität. Durch die Verwendung einer [App Service-Umgebung](#) kann Ihr Unternehmen Sicherheit und Isolation erhalten für Webanwendungen erhalten und ein virtuelles Netzwerk zur Steuerung des Datenverkehrs nutzen. Eine Option, die nur einen geöffneten Port benötigt und einen Großteil des Datenverkehrs blockiert, ist ein Produkt wie [Barracuda Web Application Firewall](#), das Ihre App Service-Umgebung schützt. Die Verbindung zu on-premises Ressourcen kann mit einem Site-to-Site-VPN oder mittels [Azure ExpressRoute](#) erfolgen.

Das „Wie“

Eine Option zum Hosten Ihrer Anwendung in Azure in einer isolierten Weise besteht darin, die vorhandene Webanwendung und Datenbank in einer Azure Virtual Machine bereitzustellen, die von einem [Azure Virtual Network](#) umgeben ist. Dies lässt sich sehr schnell bereitstellen, aber würde es auch erforderlich machen, die virtuellen Maschinen zu patchen und zu unterhalten, was Sie wahrscheinlich vor die gleichen Probleme stellt, die Sie heute bereits haben.

Verwendete Services

- *Azure Web-Apps*
- *Azure App Service-Umgebung*
- *Firewall eines Drittanbieters*
- *Virtuelles Netzwerk*
- *Optional: ExpressRoute*

Eine andere Option stellt die Nutzung einer App Service-Umgebung dar, die von einem Azure Virtual Network umgeben ist. Öffnen Sie einen einzigen Port für die Öffentlichkeit und verwenden Sie ein Produkt wie die Barracuda Web Application Firewall, die vor der App Service-Umgebung platziert wird, sodass nur genehmigte Benutzer auf die Anwendungen in dieser Umgebung zugreifen können. Diese Option verleiht Ihnen größte Flexibilität zum Scale-up oder Scale-down von Ressourcen, die in einer isolierten Umgebung abgeriegelt sind. Dabei können Sie von den Vorteilen der reichhaltigen Features von App Service-Anwendungen profitieren.

Isolieren einer Webanwendung in der Cloud

Eine App Service-Umgebung ist eine Premium Service Plan-Option von Azure App Service zur Bereitstellung einer vollständig isolierten und dedizierten Umgebung. App Service-Umgebungen sind voneinander isoliert, um nur eine einzelne Kundenanwendung auszuführen, und werden immer in einem Azure Virtual Network bereitgestellt. Auf hohem Niveau dargestellt, besteht eine App Service-Umgebung aus Compute-Ressourcen, die im dem Azure Hosted Service ausgeführt werden, ebenso wie aus Speicher, Datenbank und einem virtuellen Netzwerk, sowie aus einem Subnetz, in dem der gehostete Service läuft.

Um eine Azure-Webanwendung in einer App Service-Umgebung zu hosten, erstellen Sie zuerst die App Service-Umgebung und fügen Sie dieser dann die Webanwendung hinzu.

- [Erstellen Sie eine App Service-Umgebung](#) und wählen Sie ein Virtual Network aus, skalieren Sie Frontend und Worker-Pools, und legen Sie den Instanzenzähler sowie die Anzahl der zu verwendenden IP-Adressen fest.
- [Richten Sie Benachrichtigungen ein](#) zur Überwachung von CPU und Arbeitsspeicher für die App Service-Umgebung.
- [Fügen Sie IP-Adressen hinzu oder entfernen Sie diese](#) zu bzw. aus Ihrer App Service-Umgebung, die Ihre Anwendungen verwenden sollen.
- [Spezifizieren Sie die automatische Skalierung](#) einzelner Worker-Pools basierend auf Metriken oder Zeitplänen.

Gängige Szenarien für Cloud-Anwendungen

- [Erstellen Sie eine neue Azure Web App](#) und fügen Sie diese zu einem bestehenden App Service-Plan hinzu oder erstellen Sie in der App Service-Umgebung einen neuen Plan.
- [Falls Sie einen App Service-Plan erstellen](#), wählen Sie die App Service-Umgebung und einen Worker-Pool.

Absicherung einer Webanwendung in der Cloud

Zum Schutz Ihrer Webanwendungen können Sie eine Web-Application-Firewall (WAF) eines Drittanbieters verwenden, die eingehenden Webdatenverkehr inspiziert, um bösartige Anfragen zu blockieren. Viele dieser Firewalls kontrollieren auch die Antworten von Backend-Webservern zur Verhinderung von Datenverlust (Data Loss Prevention, DLP). In Kombination mit der Isolierung sowie der zusätzlichen Skalierung von App Services-Umgebungen bietet das eine ideale Umgebung für den Schutz von Anwendungen. Die WAF wird gehostet in einer Azure Virtual Machine. Sie können überlegen, ob Sie mindestens zwei Instanzen der virtuellen Maschine bereitstellen, um Redundanz zu erzielen und dadurch einen möglichen Single-Point-of-Failure an dieser Stelle zu beseitigen.

- [Erstellen Sie die Barracuda Web Application Firewall und stellen diese bereit](#) (oder eine ähnliche Firewall) und weisen dieser eine statische IP-Adresse zu.
- [Fügen Sie HTTP- und HTTPS-Endpunkt](#), die von Ihren Azure-Webanwendungen genutzt werden.
- [Konfigurieren Sie Ihre Firewall](#) über das Management-Portal (TCP-Port 8000 hinzufügen oder entfernen).
- [Konfigurieren Sie die Firewall](#) für den Schutz Ihrer Webanwendung.
- [Erstellen Sie eine Netzwerksicherheitsgruppe](#) und weisen diese einem Subnetz in Ihrem Azure Virtual Network zu, um den Datenverkehr von der WAF zur App Service-Umgebung einfach durch die Verwendung der virtuellen IP-Adresse (VIP-Adresse) zu beschränken.

Profi-Tipp

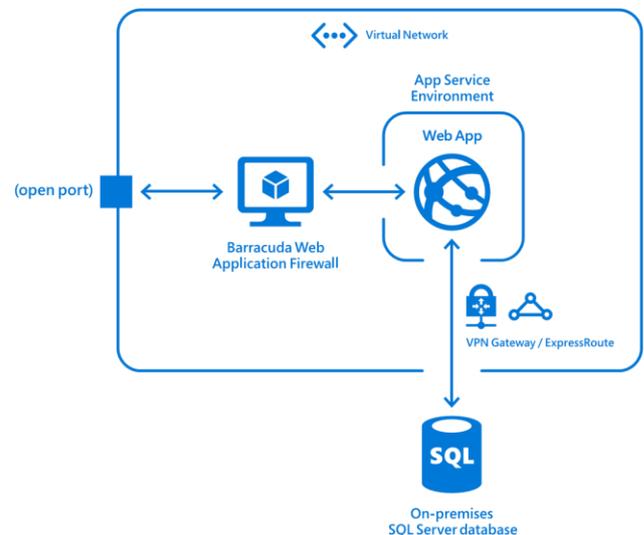
Standortübergreifend können Sie Site-to-Site-, Point-to-Site-, VNet-to-VNet-, Multi-Site- und ExpressRoute-Verbindungen verwenden. Um mehr über die Unterschiede zwischen diesen Verbindungsarten zu erfahren, besuchen Sie [diesen Link](#).

Auf on-premises Ressourcen sicher zugreifen

Für den Zugriff auf on-premises Ressourcen können Sie entweder ein Site-to-Site-VPN zwischen dem die App Service-Umgebung umgebende Azure Virtual Network und der Firewall verwenden, oder Sie fügen dem virtuellen Netzwerk Azure ExpressRoute zum Datentransfer zwischen Ressourcen über eine schnellere Verbindung hinzu.

- [Bei Verwendung eines Site-to-Site-VPN](#) fügen Sie ein Gateway-Subnetz zum Azure Virtual Network hinzu. Sodann fügen Sie Ihre lokale Site hinzu, beantragen eine öffentliche IP-Adresse für das Gateway, erstellen die Gateway-IP-Adressierungskonfiguration sowie das Gateway, konfigurieren Ihr VPN-Gerät und erstellen die VPN-Verbindung.
- Bei Verwendung von ExpressRoute [erstellen und ändern Sie eine Verbindung](#) sowie die [Routing-Konfiguration](#). Danach [verknüpfen](#) und [konfigurieren](#) Sie ein virtuelles Netzwerk für ExpressRoute.

Architekturdiagramm



Weiterführende Ressourcen

[App Service-Umgebung im Überblick](#) →

[Barracuda Web Application Firewall im Azure Marketplace](#) →

[Übersicht über die Netzwerkarchitektur von App Service-Umgebungen](#) →

Gängige Szenarien für Cloud-Anwendungen



Szenario 2: Backend für eine mobile Anwendung erstellen

In diesem Abschnitt erfahren Sie, wie Sie die Funktionalität Ihrer mobilen Anwendung um Backend-Features erweitern und das Benutzerverhalten analysieren können.

A. Ihrer Webanwendung mobile Features hinzuzufügen

Übersicht

Sie verfügen über eine bestehende Azure-Webanwendung, die mit einem Speicher-Backend verbunden ist. Ihre Kunden erwarten ein mobiles Erlebnis, sodass Sie eine mobile Anwendung erstellen. Des Weiteren möchten Sie gerne, dass die mobile Anwendung Daten und APIs mit der bestehenden Webanwendung teilt. In der Lage zu sein, Benachrichtigungen an Benutzer zu senden, würde die Benutzerbindung verbessern.

Das Problem

Die Erstellung mobiler Client-Anwendungen, die mehrere Plattformen unterstützen sollen, kann zeitintensiv sein. Entwickler müssen sich vertraut machen mit verschiedenen Programmiersprachen, Plattformen und IDEs wie zum Beispiel Xcode und Eclipse. Sie wünschen eine allgemeine Vorgehensweise, um auch mobile Anwendungen um Backend-Features wie Push-Benachrichtigungen, Offline-Datensynchronisation und automatische Skalierung zu ergänzen. Die Erstellung einer Lösung, um die API Ihrer Webanwendungen verfügbar zu machen und diese in nativen mobilen Anwendungen wiederzuverwenden, ist keine triviale Aufgabe.

Die Lösung

Microsoft bietet umfassende Lösungen sowohl für die Front-End- als auch die Cloud-fähige Backend-Entwicklung von plattformübergreifenden mobilen Anwendungen. Sie können native mobile Anwendungs-Frontends in Objective-C, Swift und Java erstellen. Als .NET-Entwickler können Sie [Xamarin](#) verwenden, um mobile Client-Anwendungen in C# zu erstellen und Ihre

Client-Geschäftslogik über iOS, Android und Windows Phone hinweg teilen.

Wenn Sie mobile Anwendungen erstellen, müssen Sie über die Client-Anwendungsfunktionalität sowie das Benutzererlebnis nachdenken – und darüber hinaus gemeinsame Backend-Features wie Push-Benachrichtigungen hinzufügen. Diese vorgefertigten mobilen und weitere Features können Sie über [Mobile Anwendungen im App Service](#) hinzufügen. Haken Sie sich in [Azure Notification Hubs](#) ein, um personalisierte Push-Benachrichtigungen an Benutzer zu senden. Danach teilen Sie Daten mit Ihrer Web-Anwendung und nutzen diese Backend-Features für eine komplette mobile Lösung.

Erstellen Sie eine neue API für Ihre Web-Anwendungen oder, falls Sie eine bestehende API haben, entnehmen Sie diese und stellen diese als [API-App in App Service](#) bereit. Dadurch können Sie die API zwischen Anwendungen teilen, die zum selben App Service-Plan gehören.

Das „Wie“

Erstellen Sie eine mobile Anwendung, möchten Sie von Services wie der automatischen Skalierung und Authentifizierung profitieren. Ein effizienter Ansatz dafür ist es, Ihre mobile Client-Anwendung zu erstellen und diese mit einer mobilen Anwendung im App Service-Backend zu verbinden, zusammen mit der bei Einbindung von Notification Hubs für Push-Benachrichtigungen. Stellen Sie Ihre API in ihrer vorliegenden Form für API-Apps in demselben App Service-Plan bereit, um diese API sowie Daten zwischen beiden Anwendungen zu teilen.

Erstellung einer mobilen Anwendung

Azure Mobile Apps ist so konzipiert, dass Sie hochgradig skalierbare und global verfügbare mobile Anwendungs-Backends erstellen können. Azure Mobile Apps unterstützt native Client-Anwendungen sowie plattformübergreifende Xamarin-native und Cordova hybride mobile Client-Anwendungen.

- Erstellen Sie eine mobile Client-Anwendung (hier finden Sie ein [Xamarin-Beispiel](#)) und verbinden diese mit Ihrem Azure Mobile App-Backend.

Gängige Szenarien für Cloud-Anwendungen

- Fügen Sie die [Synchronisierung von Offlinedaten](#) hinzu, die es Benutzern gestattet, mit der mobilen Anwendung auch ohne Netzwerkverbindung zu interagieren.

Verwendete Services

- *Azure Mobile Apps*
- *Azure API-Apps*
- *Xamarin*
- *Azure Notification Hubs*

- Im Mobilbereich bei Einstellungen wählen Sie das Speicherkonto aus, das von Ihrer Web-Anwendung verwendet wird, um die Datenverbindung hinzuzufügen.
- Erweitern Sie Ihre mobilen Anwendungen um automatische Skalierungseigenschaften, die Authentifizierung (etwa mit Active Directory), Web-Jobs, das Routing von Datenverkehr und kundenspezifische Domains.

Push-Benachrichtigungen an Benutzer senden

Azure Notification Hubs gestattet es Ihnen, personalisierte mobile Push-Benachrichtigungen plattformübergreifend mit einem einzigen API-Aufruf zu senden. Entwickler können Benachrichtigungs-Hubs in Anwendungen leicht integrieren. So erstellen Sie einen Benachrichtigungs-Hub:

- Erstellen Sie einen neuen Benachrichtigungs-Hub und [konfigurieren Sie das mobile Backend, um auf diesen Hub zu verweisen](#).
- Registrieren Sie Ihre Anwendung, um Push-Benachrichtigungen zu verwenden, und erstellen Sie ein SSL-Zertifikat.
- [Aktualisieren Sie das Server-Projekt](#), um Push-Benachrichtigungen zu senden, und [fügen Sie der Anwendung Push-Benachrichtigungen hinzu](#).

Eine API in der Cloud hosten

Wenn Sie über bestehende APIs verfügen, stellen Sie

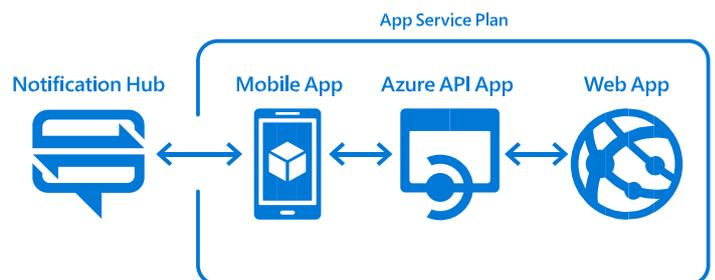
Profi-Tipp

Erfahren Sie in diesem Channel 9-Video, wie Sie [in zwei Minuten Ihre mobile Anwendung mit Azure verbinden](#).

diese in ihrer vorliegenden Form bei API-Apps in App Service bereit. Auf diese Weise profitieren Sie von Sicherheit auf Unternehmensniveau, leichter Zugriffskontrolle, hybriden Verbindungen, automatischer SDK-Generierung sowie der Integration mit Tools wie Visual Studio. Durch die Unterstützung für [Swagger API-Metadaten](#) können Sie die APIs über eine Vielzahl von Clients verwenden und automatisch Client-Code generieren. Benötigen Sie zusätzliche Funktionen für die Zugriffsverwaltung, nutzen Sie das [Azure API Management](#), um den Zugriff von Clients auf die von App Service gehosteten APIs zu steuern.

- [Laden Sie das neueste Azure SDK herunter](#) für Ihre Sprache sowie das Tool Ihrer Wahl.
- Installieren Sie das [Swashbuckle-Package](#), um mit Swagger API-Metadaten zu arbeiten.
- [Veröffentlichen Sie Ihre API](#) (Visual Studio-Beispiel) in einer neuen API-App in demselben App Service-Plan wie die bestehende Webanwendung.
- [Generieren Sie REST API-Client-Code](#) über die Swagger URL.

Architekturdiagramm



Weiterführende Ressourcen

[Ich verwende Mobile Services: Welche Vorteile bietet App Service?](#) 

[Anleitung für eine unternehmensbezogene Push-Architektur für Benachrichtigungs-Hubs](#) 

[Erste Schritte mit API-Apps und ASP.NET in Azure App Service](#) 

Gängige Szenarien für Cloud-Anwendungen

B. Benutzerverhalten analysieren mit Ihrem mobilen Backend

Sobald Sie ein mobiles Backend erstellt haben, müssen Sie wissen, was sich verbessern lässt, damit Ihre Benutzer zufrieden sind. Sie möchten aktuelles Feedback schnell erhalten mittels Echtzeitanalysen des Benutzerverhaltens und gezielter Push-Benachrichtigungen für Feature-Anfragen. Ebenso möchten Sie vielleicht Tweets über Ihr Produkt in Ihrem internen #UserFeedback Slack-Channel posten, um den Kurs schnell zu ändern und Ihre Anwendungen basierend auf dem Feedback der Benutzer zu verbessern.

Die Lösung

Verbinden Sie [Azure Mobile Engagement](#) mit einem [Azure Mobile App](#)-Backend, um Benutzerverhaltensanalysen in Echtzeit zu sammeln und personalisierte Benachrichtigungen zu senden. Erstellen Sie eine [Logik-App](#) mit einem benutzerfreundlichen Designer, der nach bestimmten Tweet-Inhalten sucht und über ein Conditional-Statement eine Nachricht an Slack sendet.

Das „Wie“

Sie können sich dafür entscheiden, Ihre eigenen Tools und Utilities zu erstellen, um Analysen für Ihre mobilen Anwendungen zu liefern. Genauso können Sie benutzerdefinierten Code für die Verbindung mit externen Ressourcen wie Twitter und Slack-APIs verwenden. Anstatt diese Lösungen selbst zu kreieren, können Sie auch auf bestehende Lösungen in Azure zurückgreifen – und sich somit darauf konzentrieren, das zu verbessern, was Ihre Anwendung so einzigartig macht.

Profi-Tipp

Tauchen Sie [tief ein](#) in die durchgängigen Ende-zu-Ende-Plattform und Tools von Microsoft zur Erstellung mobiler Unternehmensanwendungen.

Fügen Sie Ihrer mobilen Anwendung Azure Mobile Engagement hinzu, um Analysen zu sammeln und personalisierte Benachrichtigungen zu senden. Verwenden Sie Logik-Apps in demselben App Service, um eine „wenn-dies-dann-das“-Funktionalität für Tweets zu

erstellen und Nachrichten in einem Slack-Channel zu posten.

Das Backend Ihrer mobilen Anwendung hosten

Verwenden Sie Azure Mobile Apps, um das Backend für Ihre mobilen Anwendungen in Azure zu hosten.

- [Erstellen Sie eine neue mobile Anwendung](#) in App Service.
- [Konfigurieren Sie das Server-Projekt](#).

Benutzer besser einbinden durch Analysen

Mobile Engagement ist eine Software-as-a-Service (SaaS)-Plattform zur Einbindung von Benutzern, die datengesteuerte Einblicke in die Anwendungsnutzung und Echtzeit-Benutzersegmentierung bietet sowie kontextorientierte Push-Benachrichtigungen und In-App-Messaging erlaubt. Beispielsweise können Sie benutzerdefinierte Dashboards erstellen, um Key Performance-Indikatoren (KPIs) zu messen, Nutzungseingänge in einem Benutzerpfad schnell zu finden und zu beheben sowie die Benutzerzurückhaltung und -bindung nachzuverfolgen und somit festzustellen, welche Kampagnen den höchsten Return-on-Investment (ROI) einbringen. Mobile Engagement bietet In-App-Messaging-Möglichkeiten und arbeitet nahtlos mit nativen Push-Benachrichtigungsgateways wie Google GCM, Apple APNS und Microsoft MPNS zusammen.

- [Erstellen Sie eine neue Mobile Engagement-Anwendung](#).
- Kopieren Sie die Verbindungszeichenfolge für die Mobile Engagement-Anwendung und [verbinden Sie Ihre mobile Anwendung](#) damit.
- [Aktivieren](#) und [verbinden](#) Sie Ihre mobile Anwendung mit der Echtzeit-Überwachung.
- [Ermöglichen Sie Push-Benachrichtigungen](#) für die mobile Anwendung mittels Mobile Engagement.

Datenquellen verbinden für Ereignisse

Logik-Apps erlauben Entwicklern die Gestaltung von Workflows, die mit einem Auslöser beginnen und dann eine Reihe von Schritten ausführen. Jeder Schritt ruft eine API auf. Gleichzeitig wird für die Authentifizierung,

Gängige Szenarien für Cloud-Anwendungen

Sicherheit, die Erstellung von Prüfpunkten und eine robuste Ausführung Sorge getragen.

- Erstellen Sie eine neue Logik-App in demselben App Service wie Ihre mobile Anwendung.
- [Erstellen Sie eine neue Aktion](#) zur Abfrage von Tweet-Inhalten, die sich auf Ihr Produkt beziehen, und melden Sie sich bei Twitter an.
- Erstellen Sie eine bedingte Anweisung dafür, wenn der Tweet-Text Fehler enthält, um dann eine Aktion auszuführen.
- Erstellen Sie eine neue Aktion in der bedingten Anweisung, um für Ihr Team in einem Slack-Channel zu posten.

Verwendete Services

- *Azure Mobile Apps*
- *Azure Mobile Engagement*
- *Azure Logik-Apps*

Architekturdiagramm



Weiterführende Ressourcen

[Personalisierte Benachrichtigungen senden mit Mobile Engagement](#)



[Liste verwalteter APIs in Logik-Apps](#)



Gängige Szenarien für Cloud-Anwendungen

Szenario 3: Erstellung eine Internet of Things (IoT) -Anwendung

Übersicht

Sie entwickeln Anwendungen, die große Mengen an Informationen von verbundenen Geräten sammeln. Sie möchten die Nutzung und Anomalien überwachen, indem Sie Echtzeitdaten an Dashboards für Geschäfts- und Infrastrukturbenutzer senden. Ebenso möchten Sie Wachstumstrends vorhersagen und Fehler für die verbundenen Geräte durch vorausschauende Analysen vermeiden. Außerdem möchten Sie Rohdaten archivieren und diese Daten automatisch bereinigen.

Das Problem

Obwohl viele Unternehmen Analysen als Teil ihres Geschäftsmodells verwenden, kann es viele nützliche Daten geben, die innerhalb von Unternehmen zunichtegemacht werden. Die Aufgabe, viele Daten zu verarbeiten, insbesondere IoT-Daten, klingt eher abschreckend. Denken Entwickler an Machine Learning, glauben sie häufig, R-Programmierung verwenden und vorher erst einmal einen höheren Abschluss in Datenwissenschaften erwerben zu müssen.

Die Lösung

Sie denken, Sie müssen R verstehen und benötigen einen Dokortitel, um mit Azure Machine Learning zu arbeiten? Beides schadet sicherlich nicht, ist aber keineswegs erforderlich, denn: Microsoft hat Azure Machine Learning bewusst Entwickler-freundlich entworfen. Daher ist sich dieses Produkt sowohl für neugierige Anfänger geeignet als auch leistungsfähig genug für Datenwissenschaftler.

Verschiedene Azure-Technologien können Sie miteinander verbinden, um Daten aufzunehmen, vorherzusagen und an ein Dashboard zu übergeben. Nutzen Sie [Azure IoT Hub](#) und [Azure Stream Analytics](#) für Verbindungen mit ihren Geräten, greifen Sie darauf zu und verarbeiten diese Ereignisse. Speichern Sie diese Daten in [Azure Storage](#) zu Archivierungszwecken, während Sie [Azure Machine Learning](#) dazu verwenden,

Modelle für Vorhersagen zu kreieren. Verbinden Sie [Power BI](#) direkt mit Stream Analytics und schauen Sie sich diese Vorhersagen in einem Dashboard in Echtzeit für Ihre Geschäftsbenutzer an. Nutzen Sie [Azure Functions](#) zur Bereinigung der Daten, indem Sie auf Änderungen in Azure Storage reagieren.

Das „Wie“

Wettbewerbsfähige Unternehmen sind in der Lage, Daten, die sie bereits haben, zu nehmen und diese in wertvolle Vorhersagen umzuwandeln. Solche Unternehmen können die Prognosen zur Erstellung von Produktempfehlungen, Betrugserkennung sowie zur vorausschauenden Wartung verwenden. Doch wie gelangen Sie vom Punkt A, bei dem nur Rohdaten vorliegen, zum Punkt Z durch die Erstellung von Predictive Analytics-Lösungen auf Basis Ihrer Daten?

Verwenden Sie Azure IoT Hub, um pro Sekunde Millionen an Ereignissen von verbundenen Geräten zu protokollieren. Implementieren Sie Stream Analytics, um solche Ereignisse aufzunehmen und in Echtzeit zu verarbeiten. Fügen Sie Machine Learning-Funktionen hinzu, um ein Machine Learning-Modell zu erstellen und übergeben die Stream Analytics-Jobergebnisse an Azure Blob-Speicher zur Archivierung. Dann fügen Sie Ihrem Stream Analytics-Job eine Power BI-Ausgabe hinzu und erstellen ein Dashboard aus dem Datensatz. Legen Sie eine Funktion an, um Datenbereinigungsaufgaben auszuführen, wenn der Blob-Speicher einen bestimmten Größen-Schwellenwert erreicht.

Daten von verbundenen Geräten sammeln

IoT Hub ist ein neuer Service, den wir eingeführt haben, um die Bedürfnisse von IoT zu erfüllen. Der IoT Hub ist das Kernstück einer Azure IoT-Lösung und dient als das Cloud-Gateway für Ihre „Dinge“, mit dem sie sich verbinden. IoT Hub kann für Millionen von Verbindungen pro Hub skalieren und riesige Datenmengen verarbeiten. Ebenso bietet der Service die Möglichkeit, aus der Cloud zurück mit den Geräten zu kommunizieren, die er kennt, was somit Befehls- und Steuerungsfunktionen ermöglicht. Der IoT Hub beherrscht mehrere Protokolle, einschließlich HTTP, MQTT und AMPQ. Des Weiteren spielt der IoT Hub

Gängige Szenarien für Cloud-Anwendungen

eine wichtige Rolle für die Sicherheit Ihrer Lösung, indem er für jedes Gerät eine Authentifizierung unterstützt.

- Beginnen Sie mit unserer [Schritt-für-Schritt-Anleitung](#) zur Verbindung Ihrer IoT-Geräte.
- Fügen Sie Regeln mit Senden sowie Verwalten-, Senden- und Abhör-Rechten hinzu, und kopieren Sie die beiden Verbindungszeichenfolgen.

Daten in Echtzeit verarbeiten

Stream Analytics ist eine vollständig verwaltete, kosteneffiziente und Echtzeit-Engine zur Ereignisverarbeitung, die dabei hilft, tiefe Einblicke auf Daten zu gewinnen. Stream Analytics macht es einfach, Echtzeit-Analyseberechnungen für Datenstreams von Geräten, Sensoren, Websites, Infrastruktursystemen und dergleichen mehr einzurichten. Stream Analytics verbindet sich direkt mit IoT Hub zur Stream-Aufnahme. Ergebnisse können aus Stream Analytics in Speicher-Blobs oder Tabellen geschrieben werden, neben anderen Lösungen in Azure.

Optional können Sie Ergebnisse auch direkt in Power BI ausgeben.

- [Erstellen Sie einen neuen Stream Analytics-Job](#).
- [Spezifizieren Sie den Job-Input](#) für den Stream Analytics-Job.
- [Spezifizieren Sie die Job-Abfrage](#) (oder Abfragen) zur Beschreibung der Transformationen für die Echtzeit-Verarbeitung.
- [Spezifizieren Sie das Job-Ergebnis](#) des Speicher-Blob-Containers, um die verarbeiteten Daten zu speichern.
- Spezifizieren Sie ein weiteres Job-Ergebnis für Power BI sowie zum Versand von Echtzeit-Daten an ein Dashboard.

Speicherung verarbeiteter Ergebnisdaten

Tabellen oder Blob-Speicher sind verfügbar für die Speicherung der von Stream Analytics verarbeiteten Daten. Speicher ist dabei die wesentlich kostengünstigere Lösung zur Speicherung großer Datenmengen im

Vergleich zu einem traditionellen relationalen Datenspeicher.

- Erstellen Sie ein eindeutiges Speicherkonto im Portal.
- Erstellen Sie einen neuen Container und legen Sie den Zugriff darauf als Public-Blob fest.
- [Erstellen Sie gemeinsame Zugriffssignaturen](#), um Speicher-Container abzuriegeln.
- Übertragen Sie das Job-Ergebnis von Stream Analytics an den Speicher-Container.

Erstellung vorausschauender Lösungen anhand von Daten

Azure Machine Learning bietet nicht nur Tools zur Modellierung vorausschauender Analysen, sondern auch einen vollständig verwalteten Service, den Sie dazu nutzen können, Ihre vorausschauenden Modelle als gebrauchsfertige Webservices bereitzustellen. Sie können entweder Machine Learning als eine Funktion in Stream Analytics (das Ergebnisse in Blob-Speicher ablegt) oder das [Machine Learning REST API](#) verwenden und dieses mit dem Power BI REST API verbinden.

- Erstellen Sie ein neues Experiment in Azure Machine Learning Studio.
- [Erstellen Sie ein Modell](#), indem Sie Speicherdaten erhalten, Daten vorverarbeiten und Features definieren.
- Trainieren Sie das Modell durch die Anwendung eines lernenden Algorithmus.
- Bewerten und testen Sie das Modell zur Erstellung vorausschauender Lösungen.

Anzeige Ihrer Vorhersagen in einem Dashboard

Microsoft Power BI hilft Ihnen bei Verbindungen mit mehreren Datensätzen, um alle Ihre relevanten Daten an einem Ort zusammenzubringen. Power BI-Dashboards zeigen Kacheln an, die Sie anklicken können, um Reports zur weiteren Untersuchung zu öffnen. Verwenden Sie das [REST API for Power BI](#) zur Verbindung mit anderen REST-APIs und wandeln Sie Daten auf Ihrer Maschine mit [Power BI Desktop](#) um.

Gängige Szenarien für Cloud-Anwendungen

- [Fügen Sie das Job-Ergebnis](#) für den Stream Analytics-Job zu Power BI hinzu und autorisieren Sie die Verbindung .
- Fragen Sie die Daten ab, die Sie in einem Report oder Dashboard haben wollen (falls Sie einen neuen Job verwenden).
- [Erstellen Sie das Dashboard](#) in Power BI online.

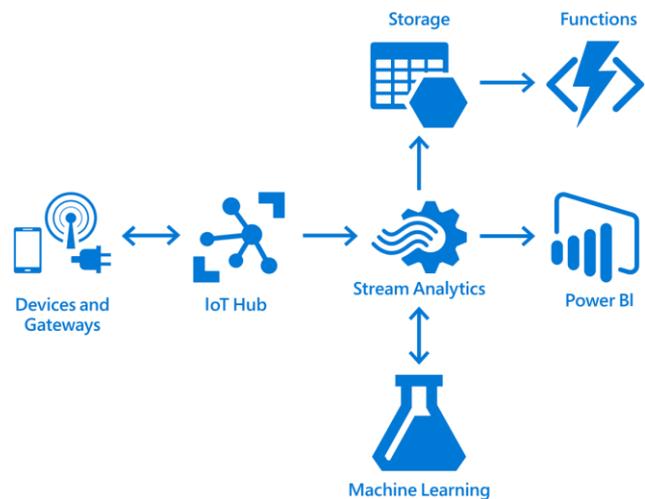
Skalierbare ereignisgesteuerte Funktionen ausführen

Azure Functions ist ein neuer Serverless-Compute-Service, der Ihnen die Erstellung einer Funktion in einer Vielzahl von Programmiersprachen erlaubt. Dadurch können Sie auf Ereignisse wie HTTP-Anfragen oder Azure Services-Ereignisse wie Azure Blob-Speicheraktualisierungen reagieren. Ihre Azure Functions laufen in einer dynamischen Compute-Umgebung, die automatisch ein bedarfsgerechtes Scale-up vornimmt. Ihnen wird lediglich die Zeit berechnet, während der Ihre Funktion läuft. [Erfahren Sie mehr über die Verwendung von Azure Functions.](#)

Profi-Tipp

Erfahren Sie in diesem Tutorial, wie Sie [Stimmungsanalysen mit Stream Analytics und Machine Learning](#) ausführen.

Architekturdiagramm



Weiterführende Ressourcen

[Azure IoT Developer Center](#)



[Machine Learning – Cheat Sheet für Algorithmen](#)



[Power BI – Übersicht und Lernen](#)



Gängige Szenarien für Cloud-Anwendungen



Szenario 4: Erstellen einer maßgeschneiderten Microservice-basierten Anwendung

Übersicht

Ihr Startup-, ISV- oder großes Unternehmen sieht sich mit explosivem Wachstum konfrontiert. Sie wollen die Gewissheit haben, dass Schlüssel-Anwendungen und Services, die das Geschäft unterstützen, skalieren und rund um die Uhr verfügbar sind, während Ihre Entwickler kontinuierlich den Code patchen und neue Features hinzufügen. Sie möchten neue Anwendungen und Services mit Cloud-nativen Architekturen erstellen, die optimiert sind für die dynamische, unbegrenzte Skalierung der Cloud.

Passen die Anwendungsbedürfnisse Ihres Unternehmens nicht zu den zuvor erwähnten normalen Vorgehensweisen, ist ein maßgeschneiderter Ansatz gefragt, um eine spezifische Lösung zu erstellen, die Ihren geschäftlichen Erfolg vorantreibt. Sind Sie ein Startup, ein ISV oder ein Großunternehmen, dessen Kerngeschäft davon abhängt, dass ein bestimmter Service verfügbar ist? Dann wollen Sie ein höheres Maß an Kontrolle und Anpassungen haben, um den Erfolg sicherzustellen.

Das Problem

Gleich, ob Sie Finanzhandelssysteme, Instant-Messaging-Services, Datensammel-Hubs oder dynamische Werbungs-Services erstellen oder Patientendaten in nationalem Maßstab verwalten, Sie wollen mehr Kontrolle und Flexibilität als die typische Web- oder mobile Anwendungsplattform bereitstellen kann.

Sie sind bereit, laufend in Entwicklung und Betrieb zu investieren einer maßgeschneiderten Architektur zu investieren, um eine Anwendung zu kreieren, die exakt mit Ihrem geschäftlichen Erfolg zusammenpasst. Allerdings wollen Sie, dass Ihre Entwickler die Erstellung auf einer Plattform durchführen, die es ihnen gestattet, sich darauf zu fokussieren, Mehrwert zu liefern in einer

agilen Weise, ohne die darunterliegende Infrastruktur verwalten zu müssen.

Die Lösung

Erstellen Sie Ihre Anwendung unter Verwendung eines Microservice-Ansatzes und stellen Sie diese auf Azure Service Fabric bereit. Die Microservice-Architektur gliedert eine Anwendung in kleine, unabhängig voneinander ausgeführte Microservices aus, was in den folgenden Vorteilen resultiert:

- Services sind für Entwickler leichter zu verstehen und zu erstellen/umzubauen.
- Services können entwickelt und bereitgestellt werden in einer unabhängigeren Weise, was die Entwicklung beschleunigt.
- Services starten und stoppen schnell. Das beschleunigt die Entwicklung und steigert die Entwicklerproduktivität.
- Es ist einfacher, nur solche Komponenten dynamisch zu skalieren, die benötigt werden, um die Nachfrage zu erfüllen.
- Das System ist toleranter gegenüber Fehlern und hat daher eine höhere Verfügbarkeit.
- Einzelne Microservices können live aktualisiert werden, ohne dass dadurch der Service unterbrochen wird.

Allerdings weisen Microservice-Architekturen einen höheren Komplexitätsgrad auf. Das macht eine PaaS-Ebene erforderlich, um sie effizient bereitzustellen und in großem Maßstab zu verwalten. Genau das erledigt die Bereitstellung auf Azure Service Fabric für Sie – sodass sich Ihre Entwickler darauf konzentrieren können, Mehrwert zu liefern.

(Einen tieferen Einblick in Azure Service Fabric und Microservices erhalten Sie im Abschnitt „[Service Fabric](#)“ später in diesem Leitfaden.)

Das „Wie“

Gestalten Sie Ihre Anwendung durch Verwendung einer Sammlung von kleineren Microservices. Jeder Service sollte:

Gängige Szenarien für Cloud-Anwendungen

- Unabhängig bereitstellbar und isoliert sein
- Seine eigenen Daten verwalten
- Mindestens einen geschützten API-Endpunkt veröffentlichen

Um den Prozess der Microservices-Erstellung einfacher zu machen, unterstützt Service Fabric das Aktoren-Programmiermodell. Dieses erlaubt es Entwicklern, objektorientierte Services und ihre Interaktionen zu modellieren zur Abstraktion von expliziten Daten- und Netzwerkoperationen, um Microservice-basierte Anwendungen auf äußerst produktive Weise zu erstellen.

Architekturdiagramm

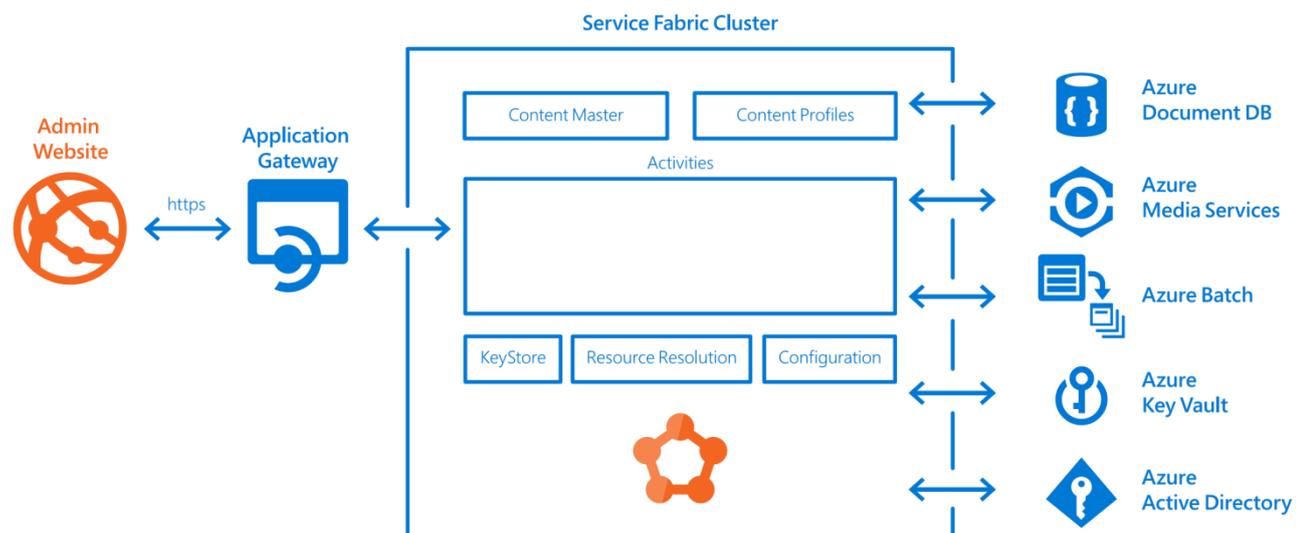
Service Fabric unterstützt sowohl Stateless- als auch Stateful-Services. Letztere verfügen über lokale Daten, die auch dann bestehen bleiben, falls der Dienst versagt. Die potenziell verschiedenen Architekturen sind grenzenlos. Als Beispiel allerdings zeigt die nebenstehende Abbildung, wie ein Kunde (TalkTalk TV) eine Microservices-Anwendung auf Service Fabric erstellte, während er sich viele andere Azure-Services wie Azure Web-Apps für die Admin-Website und Azure AD zur Authentifizierung zunutze machte.

Weiterführende Ressourcen

[TalkTalk-Kundenreferenz](#)



[Azure Service Fabric-Website](#)



Die Azure-Plattform- Services

Azure App Service

[Azure App Service](#) ist eine Cloud-Plattform zur Erstellung leistungsstarker Web- und mobiler Anwendungen für beliebige Plattformen und Geräte, die sich mit in der Cloud oder on-premises befindlichen Daten verbinden können. Geschaffen für Entwickler, ist App Service eine vollständig verwaltete Plattform mit leistungsstarken Möglichkeiten wie eingebautes DevOps, Continuous Integration in Visual Studio Team Services und GitHub, Staging- und Produktionsunterstützung sowie automatischem Patching.

App Service erlaubt es Ihnen, die folgenden Arten von Anwendungen aus einer einzigen Entwicklungsumgebung heraus zu erstellen:

- **Web-Apps:** Schnelle Erstellung und Bereitstellung geschäftskritischer Webanwendungen, die mit Ihrem Geschäft skalieren.
- **Mobile Apps:** Erstellung mobiler Anwendungs-Backends mit Benachrichtigungen, Datensynchronisierung und Authentifizierung.
- **API-Apps:** Leichte Erstellung und Verwendung von Cloud-APIs.
- **Logik-Apps:** Den Zugriff auf und die Verwendung von Daten Cloud-übergreifend automatisieren, ohne Code zu schreiben.

App Service bietet ein integriertes Set an Enterprise-Funktionen mit einer einzigen Entwicklungs- und Verwaltungsumgebung. Dadurch erhalten Sie folgende Vorteile:

- **Web- und mobile Anwendungen schnell erstellen:** Web- und mobile Backend-Anwendungen zügig erstellen, bereitstellen und verwalten für Mitarbeiter und Kunden. Nutzen Sie Ihre vorhandenen Programmiersprachenkenntnisse – .NET, Java, NodeJS, PHP oder Python. Beschleunigen Sie die Entwicklung durch den Zugriff auf eine umfangreiche Galerie von APIs, Konnektoren und Logiken, die im Azure Marketplace verfügbar sind.
- **Mit beliebigen Services verbinden und Ihre Daten freisetzen:** Dank eingebauter Konnektoren verbinden Sie Ihre Web- oder mobilen Anwendungen mit Enterprise-Systemen oder SaaS binnen Minuten. Wählen Sie aus mehr als 50 Konnektoren für Enterprise-Systeme – von SAP, Siebel und Oracle über beliebte Enterprise-SaaS-Services wie Salesforce und Office 365 bis hin zu populären Internet-Services wie Facebook, Twitter und Dropbox.
- **Leichtere Integration:** Logik-Apps lassen Sie Daten Cloud-übergreifend integrieren und Geschäftsprozesse automatisieren – binnen Minuten über eine visuelle Design-Oberfläche. Ihre Logik integrieren Sie über normale REST-APIs leicht in beliebige mobile oder Web-Anwendungen. Erstellen Sie anspruchsvolle Enterprise Application Integration B2B-Lösungen unter Verwendung von Electronic Data Interchange (EDI) und Geschäftsrichtlinien (Regelmodul).
- **Steigerung der Entwicklerproduktivität.** Optimiert für DevOps und die Continuous Integration für Visual Studio Team Services und GitHub unterstützend, können Sie sich darauf fokussieren, Ihre Anwendungen rasch zu verbessern, ohne sich Gedanken um die Infrastruktur zu machen. Anwendungsaktualisierungen

Die Azure-Plattform-Services



stellen Sie bereit mit eingebauten Funktionen für Staging, Rollback und das Testen im Produktionsbetrieb.

- **Sich auf Services auf Enterprise-Niveau verlassen:** App Service bietet Sicherheit und Verwaltung vollständig auf Enterprise-Niveau. So können Sie delegierte und rollenbasierte Administration liefern, den Datenfluss zu Ihren mobilen Anwendungen leicht absichern und verwalten sowie Ihre Assets mit eingebauten Sicherheits- und Wiederherstellungsfunktionen schützen. Die Lösung ist vollständig PCI-konform mit dedizierten Umgebungen. Außerdem besteht die Möglichkeit zur Public und Private-Clouds-übergreifenden Bereitstellung.

Als ein einzelner integrierter Service macht es App Service leicht, die obigen Anwendungsarten zu einer einzigen Lösung zusammenzustellen. Das erlaubt es Ihnen, leicht Anwendungen zu erstellen, die sich sowohl an Web- als auch mobile Clients richten. Dabei wird dasselbe Backend verwendet. Eine Integration ist sowohl mit on-premises Systemen als auch beliebten SaaS-Services möglich.

App Service-Pläne stellen ein Set von Features und Möglichkeiten dar, die Sie über Ihre Anwendungen hinweg gemeinsam verwenden können. App Service-Pläne unterstützen eine Vielzahl von Preistarifen (zum Beispiel Free, Shared, Basic, Standard und Premium), wobei jeder Tarif seine eigenen Funktionen. Auf welchem App Service-Plan eine Anwendung gehostet wird, können Sie schnell und ohne Ausfallzeit ändern.

Azure Service Fabric

Azure Service Fabric ist eine vollentwickelte, funktionsreiche Microservices-Anwendungsplattform mit eingebauter Unterstützung für Lifecycle-Management, Stateful- und Stateless-Leistung in großem Maßstab, hybride Bereitstellungen, 24x7-Verfügbarkeit und Kosteneffizienz.

Microservices sind ein Ansatz zur Anwendungsentwicklung und -bereitstellung, der sich für die Agilitäts-, Skalierungs- und Zuverlässigkeitsanforderungen moderner Cloud-

Anwendungen ideal eignet. In einem Microservices-Modell können Sie kleine, unabhängig voneinander ausgeführte Services – oder „Microservices“ – einzeln erstellen und bereitstellen. Diese arbeiten über das Netzwerk mittels veröffentlichter API-Aufrufe zusammen, um die gesamte Funktionalität der Anwendung zu liefern. Das führt zu einer feingradigen, lose gekoppelten Anwendungen, die sich zur Skalierung und Zuverlässigkeit leicht über mehrere Host-Maschinen hinweg verteilen lässt. Vergleichen Sie dies mit dem monolithischen dreistufigem Ansatz, wie er in heutigen Anwendungen üblich ist.

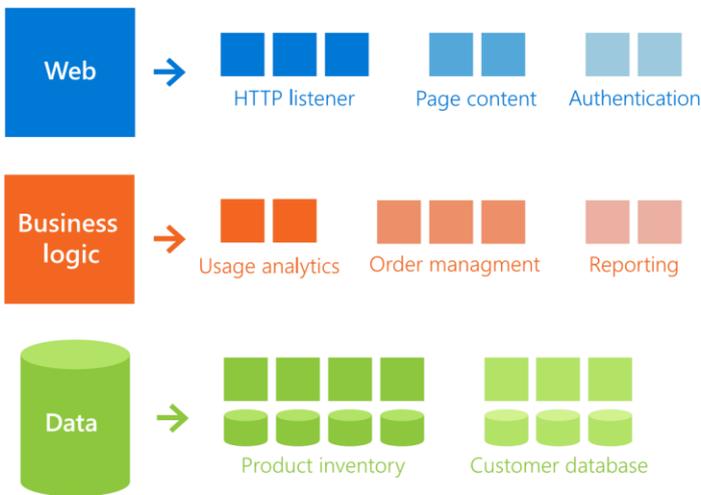
Der Begriff „Microservice“ unterstreicht die Tatsache, dass Anwendungen aus Services zusammengesetzt sein sollten, die klein genug sind, um wirklich eine Einzelrolle zu implementieren. Jeder Microservice besitzt klar definierte Contracts (API-Contracts) für andere Microservices, um miteinander zu kommunizieren und Daten zu teilen. Microservices müssen zudem in der Lage sein, unabhängig voneinander ausgeführt und aktualisiert zu werden. Diese lose Kopplung ist entscheidend zur Unterstützung einer raschen und zuverlässigen Evolution einer Anwendung. Was eine einstufige Ebene einer monolithischen Anwendung gewesen wäre, wird in viele diskrete Microservices aufgegliedert, die jeweils unabhängig und isoliert voneinander sind.

In der ersten Version läuft Azure Service Fabric auf Windows und enthält die Sprachintegration für C++ und C#. Linux- und Java-Unterstützung befinden sich bereits in der Entwicklung. Service Fabric besitzt eingebaute Unterstützung für das Lifecycle-Management, hybride Bereitstellungen und 24x7-Verfügbarkeit. Die Plattform bietet erweiterte Integritätsmodelle sowohl für die Infrastruktur als auch Microservices. Dies ermöglicht automatische integritätsbasierte Aktualisierungen und ein automatisches Rollback, was DevOps vereinfacht. Unterstützt werden sowohl Stateless- als auch Stateful-Microservices mit Leadership-Wahl zur Unterstützung der Datenkonsistenz und einem Framework zur Zustandsreplikation, das Hilfestellung bei Transaktionen für Stateful-Datengarantien leistet. Erfahren Sie mehr über den Einsatz von [Service Fabric auf Azure](#).

Die Azure-Plattform-Services



Service Fabric-Anwendungen lassen sich zusammenstellen sowohl als Stateless- als auch Stateful-Microservices. Die zuverlässige Stateful-Datenverwaltungs-Service Fabric ist in hohem Maße darauf abgestimmt, um für Daten ein Maximum an Verfügbarkeit und Dauerhaftigkeit zu erzielen, während jeder Zyklus und jede I/O-Operation für hohe Leistung optimiert wird.



Um die Microservice-Entwicklung produktiver zu gestalten, verfolgt Service Fabric einen [Aktoren-Programmiermodell](#) genannten Ansatz. Dieser ermöglicht es Ihnen, Microservices als „Aktoren“ zu betrachten, von denen jeder logische Entity-Funktionen mit Status repräsentiert. Dabei spielt es keine Rolle, ob es sich um ein IoT-Gerät, einen Videospiele, eine Spielesitzung oder Patienten und Ärzte in einem Gesundheits-Provider-Netzwerk handelt. Eine Möglichkeit, das Aktoren-Programmiermodell zu betrachten ist als objektorientierte Microservices, was die Anwendungsentwicklung erheblich vereinfachen kann.

Service Fabric ist bei Microsoft seit 2010 produktiv im Einsatz und versorgt viele unserer Anwendungen und Services, einschließlich Azure SQL-Datenbank, DocumentDB, Intune, Cortana und Skype for Business. In seinen größten Bereitstellungen werden Hunderttausende von Microservices über Tausende von Servern hinweg verwaltet. Wir haben exakt dieselbe Technologie genommen und Service Fabric direkt veröffentlicht als ein Service auf Azure. Demnächst wird die Technologie auch

für den on-premises Einsatz sowie für andere Clouds verfügbar sein.

Die Integration von Visual Studio in Service Fabric vereinfacht das Coding und die Bereitstellung. Es gibt ein vollständiges Entwicklermaschinen-Erlebnis, bei dem die eigentliche Service Fabric-Runtime auf einer einzelnen Box läuft. Zu Validierungs- und Diagnosezwecken wird dabei ein aus mehreren Servern bestehender Service Fabric-Cluster simuliert.

Schließlich lässt sich Service Fabric leicht in Ihren Anwendungslebenszyklus aufnehmen. Dies gestattet eine nahtlose Continuous Integration (CI) / Continuous Delivery (CD) sowohl von Stateless- als auch Stateful-Aktualisierungen. Um bei der Verwaltung des Anwendungslebenszyklus zu helfen, unterstützt Service Fabric sowohl Rolling- als auch Side-by-Side-Upgrades (zum Beispiel Scaling-up zu Version 2 während Scaling-down zu Version 1). Wird festgestellt, dass das Upgrade die Anwendungsintegrität verschlechtert hat, erfolgt automatisch ein Rollback der Aktualisierungen.

Azure Cloud Services

[Azure Cloud Services](#) sind für die Unterstützung von skalierbaren, zuverlässigen und günstig zu betreibenden Anwendungen konzipiert. Ein Anwendungs-Service wird auf virtuellen Maschinen gehostet, und ähnlich verhält es sich auch bei Azure Cloud Services. Allerdings haben Sie hierbei mehr Kontrolle über die virtuellen Maschinen. Sie können Ihre eigene Software auf Azure-Cloud Services virtuellen Maschinen installieren und sich mit diesen remote verbinden.

Mehr Kontrolle bedeutet auch weniger Bedienkomfort – es sei denn, Sie benötigen zusätzliche Kontrollmöglichkeiten. In der Regel ist es schneller und einfacher, eine Webanwendung bei Web-Apps in Azure App Service zum Laufen zu bringen als bei Azure Cloud Services.

Die Technologie bietet zwei leicht unterschiedliche Optionen für virtuelle Maschinen: Instanzen von *Web-Rollen* führen eine Windows Server-Variante mit IIS aus, während bei Instanzen von *Worker-Rollen* die gleiche

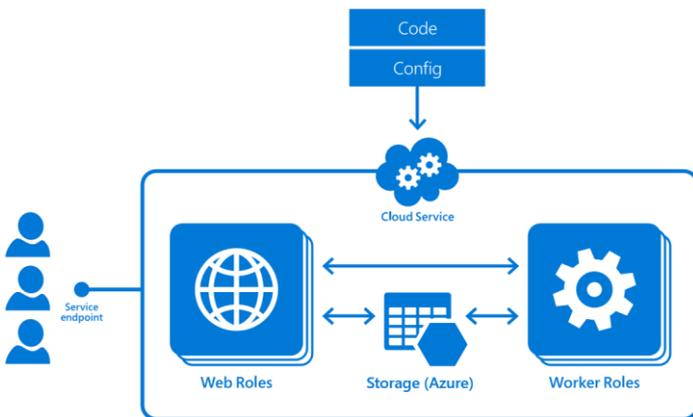
Die Azure-Plattform-Services



Windows Server-Variante ohne IIS ausgeführt wird. Eine Cloud Services-Anwendung setzt auf eine Kombination dieser beiden Möglichkeiten.

Beispielsweise könnte eine einfache Anwendung nur eine Webrolle verwenden, während eine komplexere Anwendung eine Webrolle verwenden könnte, um eingehende Anfragen von Nutzern zu behandeln, und diese Bearbeitung dieser Anfragen dann an eine zu erstellende Worker-Rolle weiterreichen würde (diese Kommunikation könnte den Service-Bus oder Azure Queues verwenden).

Auch wenn Anwendungen in virtuellen Maschinen ausgeführt werden, ist es wichtig zu verstehen, dass Azure Cloud Services PaaS bietet und nicht IaaS. Alles, was Sie machen müssen, ist Ihre Anwendung bereitstellen. Die Verwaltung der Plattform, auf der die Anwendung läuft, einschließlich der Bereitstellung neuer Versionen des Betriebssystems, wird für Sie erledigt.



über die ganze Breite von Azure hinweg generiert werden. Ob Sie auf Änderungen in Azure-Speichercontainern, in Web-Hooks unterstützenden SaaS-Produkten ausgesendete Ereignisse oder auf Aufrufe für einen HTTP-Endpunkte reagieren möchten – all das ist leicht einzurichten und erfordert nur eine minimale Konfiguration.

Ihre Funktionen können Sie in einer Vielzahl von Programmiersprachen implementieren, einschließlich JavaScript, C#, Python und PHP. Des Weiteren können Sie aus Scripting-Optionen wie Bash-Scripts, PowerShell-Scripts oder Windows-Batchdateien wählen. Azure Functions bietet ein intuitives, Web-basiertes Code-Bearbeitungserlebnis. Ebenso lassen sich Sie vorkompilierte ausführbare Dateien hochladen und triggern, die in dem Entwicklungs-Tools Ihrer Wahl erstellt wurden. Ihre Azure Functions können Sie schnell und einfach durchlaufen mit der Continuous Delivery unter Verwendung von Visual Studio Team Services, GitHub oder BitBucket. Mittels der eingebetteten können Sie Ihre Azure Functions sehr einfach überwachen und Fehlersuchen durchführen.

Azure Functions führt ein neues Preismodell ein: Ihnen wird nur die Zeit in Rechnung gestellt, wenn Ihr Code läuft. Beim Aufruf einer Azure Function wird diese mit so vielen Ressourcen versehen, wie sie zur Ausführung benötigt – solange sie ausgeführt wird. Sie können für die Absicherung Ihrer Azure Functions sorgen, indem Sie diese in einer [App Service-Umgebung](#) hosten, die so konfiguriert wird, dass sie nur von internen Netzwerken adressierbar ist.

Azure Functions

Azure Functions ist ein ereignisgesteuerter Serverless-Compute-Service. Dieser erweitert die bestehende Azure-Anwendungsplattform um Möglichkeiten zur Implementierung von Code, der von Ereignissen getriggert wird, die in anderen Azure-Services, SaaS-Produkten und on-premises Systemen auftreten.

Azure Functions vergrößert die [marktführende](#) PaaS-Plattform von Microsoft und baut auf der existierenden WebJobs-Infrastruktur auf, damit Entwickler leicht Code implementieren können, der auf Ereignisse reagiert, die

Auf IaaS aufbauend

Das Host-Betriebssystem ist *die* ursprüngliche Anwendungsplattform. Viele Entwickler schreiben weiterhin Anwendungen, die auf Windows Server, Linux oder anderen Betriebssystemen nativ laufen. Traditionell würde das Betriebssystem auf einem einzigen physischen Host-Computer aufgespielt. Die darauf installierten und laufenden Anwendungen nutzen Ressourcen und Funktionen (wie zum Beispiel Dateien, Netzwerk, Time-Slicing, Speicher und Sicherheit), die vom Betriebssystem bereitgestellt werden. Heutzutage würde das Betriebssystem sehr wahrscheinlich auf einer virtuellen Maschine ausgeführt werden anstatt auf einem physischen Host, aber das Muster bleibt dasselbe: Die Anwendung läuft auf einem einzigen Computer und konsumiert Services, die vom Betriebssystem bereitgestellt werden.

Diese Anordnung bietet Entwicklern eine sehr detaillierte Kontrolle über die Umgebung, in der ihre Anwendung ausgeführt wird. Abhängig von der Programmierumgebung gestattet diese Konstellation ein maximales Maß an Anpassungen für leistungskritische Szenarien. Jedoch ist der Entwickler oder sein IT-Mitarbeiter nun verantwortlich für die Instandhaltung und Konfiguration des Betriebssystems sowie für die komplizierten Details, um es mit der Anwendung kompatibel zu machen. Muss das Betriebssystem aktualisiert werden, ist die Anwendung auf Kompatibilität zu testen, bevor das neue System im Produktionsbetrieb live geht.

Eine gravierende Beschränkung bei dieser Art der Anwendung – was das Aufkommen von Anwendungsservern und jetzt der Cloud verdeutlicht – besteht darin, dass für die Anwendung zwar ein Scale-up erfolgen kann, indem der Host-Maschine entsprechende Kapazitäten hinzugefügt werden. Für die Anwendung aber kann kein leichtes Scale-out erfolgen, um über mehrere Maschinen hinweg ausgeführt zu werden. Dazu müssten Entwickler einen Wechsel hin zu Stateless- und mit

Lastverteilung versehenen Modellen durchführen und weitere Codeänderungen vornehmen – nur damit es zur Infrastruktur passt.

Zudem Anwendungen mit dem Betriebssystem verwoben, was der Vorgang der Migration von Anwendungen zwischen Betriebssystemversionen sehr riskant macht, ohne vollständige Tests auf der neuen Betriebssystemversion. Das Docker Packaging-Format und die zugehörigen Werkzeuge sind beliebt geworden, denn sie bieten eine Möglichkeit, portable Images von Anwendungen und Betriebssystemabhängigkeiten zu erstellen, die sich ohne das Risiko inkompatibler Abhängigkeiten leicht zwischen virtuellen Maschinen und Containern verschieben lassen.

Azure bietet Entwicklern eine Reihe von Optionen, um direkt auf dem Betriebssystem aufzubauen:

- **Virtual Machines:** Azure stellt eine breite Palette von Windows Server- und Linux-basierten Betriebssystem-Images zur Verfügung.
- **Virtual Machine Scale Sets:** Virtual Machine Scale Sets bieten einen leicht zu bedienenden „Schieberegler“-basierten Mechanismus zur Skalierung identischer Gruppen von Stateless virtuellen Maschinen.
- **Azure Container Service:** Container Service ist ein vollständig Open-Source-basierter Service. Dieser enthält Funktionen zur Bereitstellung, Orchestrierung und dem Failover von Docker-basierten Images über virtuelle Maschinen hinweg. Genutzt werden dazu Apache Mesos, Mesospheres Marathon oder Docker Swarm.
- **Azure Marketplace:** Azure Marketplace beinhaltet viele beliebte Anwendungsplattformen/PaaS-Plattformen als Virtual-Machine-Images, inklusive Pivotal's Cloud Foundry. Diese Images können Sie selbst betreiben und verwalten.

Auf IaaS aufbauend



Docker und Container

Sie können nicht über Cloud Computing diskutieren, ohne über Container und das Docker-Toolset zu sprechen. Unternehmen in allen Branchensegmenten möchten verstehen, was Container sind, was sie für Anwendungen in der Cloud bedeuten, und wie sie sich am besten verwenden lassen für ihre spezifischen Entwicklungs- sowie IT-Betriebsszenarien.

Container sind eine Form der Betriebssystem-Virtualisierung, ähnlich virtuellen Maschinen. Allerdings arbeiten Container innerhalb einer Instanz des Betriebssystems, was den Eindruck eines vollständig isolierten und unabhängigen Betriebssystems vermittelt. Für die in dem Container laufende Anwendung sehen die lokalen Festplatten wie das Original mit den Betriebssystemdateien aus. Im Arbeitsspeicher scheinen sich nur die Dateien und Daten eines frisch gebooteten Betriebssystems zu befinden, und das einzige, was läuft, ist das Betriebssystem. Das Host-Betriebssystem kontrolliert ebenfalls Ressourcen wie CPU, RAM und Netzwerkbandbreite. So wird sichergestellt, dass ein Container die Ressourcen erhält, die er erwartet und dass er keine Auswirkungen auf die Leistung anderer auf dem Host laufender Container hat.

Die Kombination aus sofortiger Inbetriebnahme, die aus der Betriebssystemvirtualisierung herrührt, und der zuverlässigen Ausführung, die sich aus der Isolation und Ressourcen-Steuerung ergibt, macht Container ideal für die Anwendungsentwicklung und das Testen, und Entwickler können das schnell wiederholen.

Da ihre Umgebung sowie die Ressourcen-Nutzung über Systeme hinweg konsistent sind, funktioniert eine containerisierte Anwendung auf einem Entwickler-System auf die gleiche Weise wie auf einem Produktionssystem. Vom sofortigen Start sowie dem kleinen Footprint profitieren auch Cloud-Szenarien durch die Möglichkeit zum schnellen Scale-Out von Anwendungen. Außerdem passen viele weitere Anwendungsinstanzen auf eine Maschine, als wenn sie jeweils in einer virtuellen Maschine wären, was die Ressourcenauslastung maximiert.

Azure Container Service

Container bieten eine kompakte Form der Virtualisierung mit kurzen Startzeiten, hocheffizienter Systemauslastung (ermöglicht Hunderte von Containern auf einem einzelnen

Computer) und, in Docker-Images, ein Verpackungsformat, das Umgebungs-übergreifend vollständig portabel ist.

Durch die Verwendung dieses gemeinsamen Image-Formats können Sie Ihre Anwendung auf einer lokalen Entwicklung-Box testen – mit dem exakt gleichen Bereitstellungs-Artefakt wie im späteren Einsatz, was die Notwendigkeit für weitere Tests auf Produktionsumgebungen verringert. Die von Containern angebotene Agilität und Skalierung stellt jedoch eine Verwaltungsherausforderung für Entwickler und IT-Administratoren bei der Bereitstellung von Anwendungen in großem Maßstab dar.

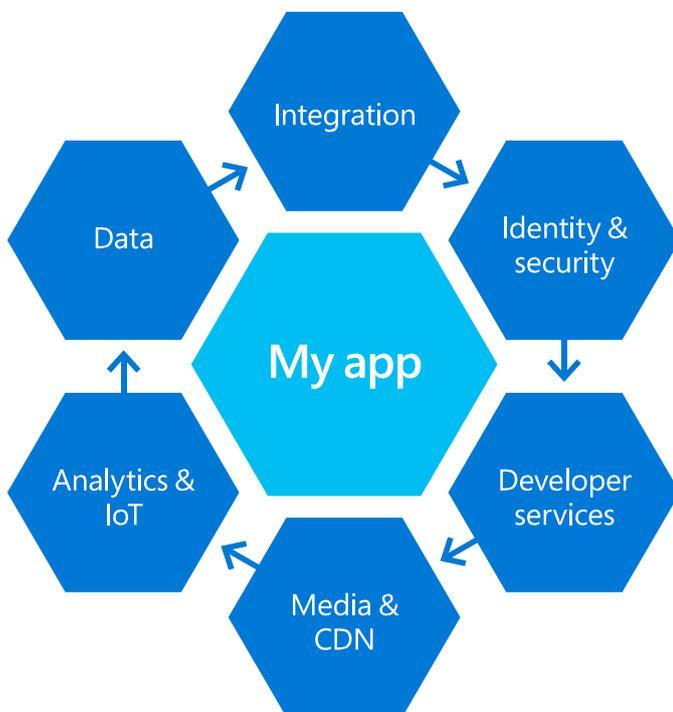
Azure Container Service bietet eine Möglichkeit zur Vereinfachung der Erstellung, Konfiguration und Verwaltung eines Clusters von virtuellen Maschinen, die vorkonfiguriert sind für die Ausführung containerisierter Anwendungen. Durch die Verwendung einer optimierten Konfiguration beliebiger Open-Source-Tools für die Zeitplanung und Orchestrierung erlaubt es Ihnen Container Service, Ihre vorhandenen Kenntnisse zu nutzen oder auf eine große und wachsende Community-Expertise zurückzugreifen, um Container-basierte Anwendungen auf Microsoft Azure bereitzustellen und zu verwalten.

Container Service verwendet das Docker-Container-Format, um sicherzustellen, dass Ihre Anwendungen vollständig portabel sind. Ebenso können Sie Marathon und Apache Mesos oder Docker Swarm nutzen und so sicherstellen, dass diese Anwendungen skalierbar sind für Tausende, oder sogar Zehntausende, von Containern.

Azure Container Service veröffentlicht normale API-Endpunkte für Ihren gewählten Orchestrator. Durch die Verwendung dieser Endpunkte können Sie von jeder Software profitieren, die imstande ist, sich mit diesen Endpunkten zu unterhalten. Im Fall des Docker Swarm-Endpunkts zum Beispiel wollen Sie vielleicht Docker Compose verwenden, während Sie bei Apache Mesos sich eventuell für die Mesosphere DCOS Befehlszeilenoberfläche entscheiden.

Zum jetzigen Zeitpunkt unterstützt der Service nur Linux-Container. Microsoft wird künftig auch [Windows Server Container, die Docker nutzen](#) unterstützen, und [Apache Mesos wird nach Windows portiert](#). Das erlaubt es uns, Azure Container Service in Zukunft um die Unterstützung für Windows Server-Container zu erweitern.

Ihrer Anwendung herausragende Merkmale hinzufügen



Nachdem Sie die Frage, was die Eigenschaften Ihrer Anwendung sind, adressiert haben, wo diese ausgeführt wird und welchen Design-Ansatz Sie ergreifen sollten, steht Ihnen die ganze Vielfalt der leistungsstarken Azure Services zur Verfügung, um Ihre Anwendung wirklich großartig zu machen.

Heute gibt es mehr als sechzig – weitere kommen hinzu – Azure Services, die Sie zur Entwicklung und zum Hosten, zur Verwaltung, Unterstützung, Absicherung, Integration und Erweiterung sowie zur Ergänzung Ihrer Anwendungsentwicklung nutzen können. In diesem Abschnitt betrachten wir einige der am häufigsten verwendeten Services, die Sie berücksichtigen sollten, wenn Sie moderne Anwendungen erstellen.

Schauen Sie sich die [komplette Liste aller Azure Services](#) an.

Ihrer Anwendung herausragende Merkmale hinzufügen



Datenbank-Services

PaaS-Datenbank-Services (auch als *Database-as-a-Service* bezeichnet) besitzen eine Reihe von Vorteilen gegenüber dem IaaS-Ansatz, bei dem ein Datenbank-Management-System (DBMS) in einer virtuellen Maschine ausgeführt wird:

- **Vollständig verwaltet:** Sie brauchen kein DBMS zu installieren, zu patchen oder zu verwalten.
- **Eingebaute Skalierbarkeit:** Anstatt Datenbank-Server manuell hinzuzufügen, wenn die Anwendungslast zunimmt, kann ein PaaS-Datenbank-Service dies nach Bedarf ohne Ausfallzeit erledigen.
- **Eingebaute Zuverlässigkeit und Fehlertoleranz:** Während das Einrichten eines geclusterten DBMS spezielle Kenntnisse und viel Zeit erfordert, kann ein PaaS-Datenbank-Service dies für Sie erledigen.
- **Service Level Agreements (SLAs):** Verlassen Sie sich auf Azure, um Ihre Verfügbarkeitsziele zu erfüllen, anstatt sich Sorgen um die Verfügbarkeit zu machen.

SQL-Datenbank

Basierend auf SQL Server bietet Azure SQL-Datenbank einen vertrauten relationalen Speicher, einschließlich der Unterstützung für SQL-Abfragen, Transaktionen über eine gesamte Datenbank hinweg und Stored-Procedures. Der Service bietet eingebaute Fehlertoleranz und Skalierbarkeit.

SQL-Datenbank ist eine gute Wahl, wenn eine Anwendung die volle Leistung eines relationalen Systems benötigt. Es ist ebenfalls gut geeignet, wenn das Entwickler-Team bereits mit SQL und relationalen Technologien vertraut ist. Da dieser PaaS-Datenbank-Service auf SQL Server basiert, ist es im Allgemeinen nicht schwierig, ihn zu lernen.

DocumentDB

Wie der Name bereits zum Ausdruck bringt, speichert Azure DocumentDB *Dokumente*, und jedes enthält JSON-Daten. Dieser PaaS-Datenbank-Service erlaubt RESTful-Zugriff auf die Dokumente, die er beinhaltet. Des Weiteren lässt er Anwendungen Abfragen unter Verwendung eines erweiterten Subset von SQL erstellen. Und wie SQL-Datenbank bietet auch DocumentDB Transaktionen, eingebaute Skalierbarkeit und eingebaute Hochverfügbarkeit.

DocumentDB ist eine attraktive Möglichkeit für Entwickler, in JavaScript und anderen modernen Sprachen zu arbeiten, von denen alle eingebaute JSON-Serializer bieten. Es eignet sich ebenfalls gut für Situationen, bei denen sich die Datenstruktur häufig ändert, da DocumentDB im Gegensatz zu einer relationalen Datenbank kein festes Schema definiert.

HDInsight HBase

HBase ist Teil der Hadoop-Technologiefamilie und für die Big Data-Verarbeitung konzipiert. HBase kann betrachtet werden als *Column-Family-Speicher*. Die Spalten in jeder Tabelle sind in Familien gruppiert, und bei Anfragen nach Daten lässt sich angeben, in welche Column-Family hineingesehen werden soll. Im Gegensatz zu relationalen Tabellen ermöglicht HBase jedoch das Hinzufügen einer neuen Spalte zu einer Column-Family während der Laufzeit – das Schema ist nicht fest. Des Weiteren wurde HBase für hohe Skalierbarkeit konzipiert, was Anwendungen Tabellen mit Spalten Millionen und Milliarden von Zeilen erstellen lässt.

HBase ist eine exzellente Wahl für Anwendungen, die große Tabellen mit geringer Dichte erstellen müssen. Es ist ebenfalls eine gute Option, wenn die gespeicherten Daten mit Hive oder einer anderen HDInsight-Technologie verarbeitet werden, da sie alle auf derselben zugrundeliegenden Clustering-Technologie aufbauen.

Azure-Tabellen

In manchen Situationen wird die volle Leistung einer relationalen Datenbank benötigt, einschließlich SQL-Abfragen gegen relationale Tabellen. Andere Situationen jedoch können mit einem viel einfacheren Ansatz zur Speicherung von und zum Zugriff auf Daten auskommen. Trotz des Namens dieses PaaS-Datenbank-Services speichert er Daten nicht wirklich in Tabellen. Stattdessen greift eine Anwendung auf Daten zu, indem sie einen eindeutigen Schlüssel bereitstellt. Dieser Service liefert zurück, was auch immer diesem Schlüssel zugeordnet ist. In vielen Situationen funktioniert dieser einfache Ansatz überraschend gut. Denken Sie zum Beispiel an die Speicherung von Benutzerprofilen: Jeder Benutzer kann einen eindeutigen Schlüssel haben, wobei jeder Schlüssel Zugriff bietet darauf, was an Profildaten für diesen Benutzer gespeichert ist. Verschiedene Benutzer können

Ihrer Anwendung herausragende Merkmale hinzufügen



unterschiedliche Daten – es gibt kein festes Schema – haben, sodass Azure-Tabellen einen flexiblen Ansatz bieten.

Andere Daten-Services

Neben den bisher beschriebenen Technologien bietet Azure zwei weitere PaaS-Datenbank-Services für die Arbeit mit operativen Daten. Keiner davon adressiert dieselben Probleme wie die bereits erwähnten Azure-Datenbank-Services, aber beide sind ebenfalls wichtig zu verstehen. Bei diesen beiden Diensten handelt es sich um Azure Search und Azure Redis Cache.

Azure Search

Für viele Menschen ist die Suche zur attraktivsten Methode zur Interaktion mit Anwendungen geworden. Warum lassen Sie die Benutzer einer Anwendung nicht danach suchen, was sie interessiert, ähnlich wie sie es im Internet machen würden, anstatt Elemente aus Menüs auswählen? Diese Möglichkeit würde viele Anwendungen leichter bedienbar machen.

Azure Search hilft Entwicklern dabei, ihre Anwendungen um Suchfunktionen zu erweitern. Dabei stehen Merkmale wie die automatische Formatierung von Suchbegriffen in Ergebnisse in Fettschrift und eine Option zur Steuerung der Reihenfolge zur Verfügung, in der diese Ergebnisse zurückgegeben werden. Anregungen unterstützt der Service ebenfalls, um mögliche Suchbegriffe basierend auf der ursprünglichen Eingabe des Benutzers bereitzustellen.

Redis Cache

Mit Redis Cache kann eine Anwendung auf Daten aus jedem Azure PaaS-Datenbank-Service wie gewohnt zugreifen. Daraufhin ist die Anwendung in der Lage, eine Kopie dieser oder anderer Daten in Redis Cache zu speichern. Benötigt die Anwendung diese Daten künftig, kann sie auf die im In-Memory-Cache befindlichen Daten zugreifen, anstatt dafür bis zum PaaS-Datenbank-Service „zurückzugehen“. Auf diese Weise erfolgen Zugriffe schneller, sodass Anwendungen bessere Reaktionszeiten haben und mehr Benutzer gleichzeitig bedienen.

(Hinweis: Der Datenbank-Service-Abschnitt wurde adaptiert vom Whitepaper „[Data in a PaaS World: A Guide for New Applications](#)“ von David Chappell.)

Erweiterte Analysen & IoT

Erweiterte Analysen helfen dabei, Daten in intelligente Handlungen zu verwandeln. Möglich machen es die Sammlung und Verwaltung einer unbegrenzten Menge an Daten, die Erweiterung von Anwendungen um Vorhersageerkenntnisse sowie die Operationalisierung von Datenwissenschafts-Pipelines zum iterativen Lernen.

Cortana Intelligence Suite

Diese vollständig verwaltete Big Data- und erweiterte Analyse-Suite ermöglicht es Ihnen, Ihre Daten in intelligente Handlungen umzuwandeln:

- **Informationsverwaltung:** Orchestrierung der Datenbewegung auf einer vollständig verwalteten, durchgängigen Ende-zu-Ende-Plattform. Verwenden Sie [Azure Data Factory](#) zum Bau von Pipelines und um Daten von Services zu sammeln und zu orchestrieren, die Sie für einfachere Analysen verwenden. Zusätzlich können Sie [Azure Data Catalog](#) zur effizienten Verwaltung von Datenquellen nutzen, während [Azure Event Hubs](#) einen Staging-Bereich für eingehende Streaming-Daten bietet.
- **Big Data-Speicher:** Speichern und verwalten Sie strukturierte Daten mit [Azure SQL Data Warehouse](#), das elastisch skaliert mit massiv paralleler Verarbeitung. Mit [Azure Data Lake-Speicher](#) implementieren Sie ein hyper-skaliertes Repository ohne Dateigrößenbeschränkungen für unstrukturierte Daten, um massiven Durchsatz und eine hohe analytische Leistung zu erzielen.
- **Machine Learning und Analysen:** Entwerfen und veröffentlichen Sie Vorhersagemodelle mit [Azure Machine Learning](#), verwenden Sie [Azure HDInsight](#) zur Analyse von Daten in Storm und Spark für Hadoop-Umgebungen, und integrieren Sie Ihren Code aus R oder Python. Mit [Azure Data Lake-Analysen](#) und [Azure Stream Analytics](#) analysieren Sie jede Art oder jede mögliche Größe von Daten, die Sie benötigen. Darüber hinaus können Sie [Microsoft Power BI](#) verwenden, um reichhaltige Visualisierungen zu erstellen, die Ihre Daten zum Leben erwecken.
- **Intelligence:** Erkunden Sie die [Cortana Intelligence-Galerie](#), um vollständig verwaltete APIs herunterzuladen

Ihrer Anwendung herausragende Merkmale hinzufügen



und darüber vorausschauende APIs für Geschäftsszenarien ([Empfehlung](#), [Prognose](#), [Anomalieerkennung](#) und Kundenabwanderung) schnell zu implementieren – ebenso wie [Cognitive Services](#)-APIs (Erkennung, Gesicht, Text und Sprache) für neue Wege bei der Interaktion mit Kunden. Beginnen Sie mit Lösungsvorlagen, die auf Ihre speziellen geschäftlichen Anforderungen zugeschnitten sind. Zusätzlich integrieren Sie Ihre Analyse-Services und -Modelle in intelligente Agenten (Bot Framework) sowie in Cortana, unseren persönlichen digitalen Assistenten, der Benutzer per Spracheingabe natürlich interagieren lässt oder diesen proaktive Benachrichtigungen sendet.

Gängige Szenarien

- **Gesundheitswesen:** Die Verbreitung von Gesundheitsdaten – von neuen Datenquellen, gestreamt in Echtzeit, bis hin zu historischen Daten, die in den Systemen von Gesundheitsanbietern gespeichert sind – in Kombination mit der Leistung erweiterter Analysen kann dabei helfen, aktuelle Herausforderungen der Gesundheitsbranche in prädiktive und präskriptive Lösungen umzusetzen.
- **Finanzdienstleistungen:** Bestehende Anti-Geldwäsche-Systeme sind robust und gut entwickelt durch jahrelange Iteration und Verbesserung, doch Unsummen gehen durch Geldwäsche noch immer jedes Jahr verloren. Mit Big Data-Tools zur Verwaltung unstrukturierter Daten erweitert die Cortana Intelligence Suite bestehende Systeme, um zusätzliche Klarheit bei der Geldwäschebekämpfung zu schaffen.
- **Einzelhandel:** Die Verbreitung strukturierter und unstrukturierter Daten kombiniert mit der Leistung erweiterter Analyselösungen bietet die Gelegenheit, Big Data-Probleme zu lösen. Die in der Cortana Intelligence Suite vereinten Technologien können Einzelhandelsunternehmen dabei helfen, Big Data erfolgreich in Einblicke in personalisierte Kundenerlebnisse zu verwandeln.

Internet of Things (IoT): Internet der Dinge

Azure IoT Suite ist eine Enterprise-Lösung, die es Entwicklern ermöglicht, schnell mit einer Reihe von erweiterbaren vorkonfigurierten Lösungen zu beginnen,

um häufige IoT-Szenarien wie die Remoteüberwachung und vorausschauende Wartung zu adressieren. Es handelt sich um vollständige, Ende-zu-Ende arbeitende Lösungen (inklusive simulierter Geräte), die von den Azure Services Gebrauch machen.

- **Azure IoT Hub:** Entwickler können neue Geräte einfach und sicher anbinden sowie bestehende Verbindungen zu bestehenden Geräten herstellen. Unter Verwendung von Open-Source-Geräte-SDKs für mehrere Plattformen, darunter Linux und Windows, lassen sich Befehle und Benachrichtigungen auf zuverlässige (über eine intermittierende Verbindung) und sichere Weise an die angeschlossenen Geräte senden sowie die Nachrichtenübermittlung verfolgen.
- **Azure Event Hubs:** Ein hochgradig skalierbarer Service für das Veröffentlichen und Abonnieren von Ereignissen, der Millionen Ereignisse pro Sekunde erfassen und an mehrere Anwendungen streamen kann. Dies ermöglicht es Entwicklern, Daten zu verarbeiten und zu analysieren, die von verbundenen Geräten und Anwendungen erzeugt werden. Anschließend lassen sich diese mit einem beliebigen Echtzeit-Analyseanbieter oder mit Batch-/Speicheradaptern transformieren und speichern.
- **Azure Stream Analytics:** Schnell können Entwickler kostengünstige Lösungen entwickeln und bereitstellen, um in Echtzeit Einblicke in Geräte, Sensoren, Infrastruktur und Anwendungen zu erhalten – etwa zur Echtzeit-Remoteverwaltung und -überwachung oder für Einblicke in Geräte wie Mobiltelefone und verbundene Autos (Connected Cars).
- **Azure Machine Learning:** Ein leistungsstarker Cloud-basierte Predictive Analytics-Service, der es ermöglicht, vorausschauende Modellen als Analyse-Lösungen schnell zu erstellen und bereitzustellen. Er bietet Tools zur Modellierung vorausschauender Analysen sowie einen vollständig verwalteten Service, um vorausschauende Modelle als gebrauchsfertige Webservices bereitzustellen. Auf diese Weise können Sie Vorhersagemodelle schnell erstellen, testen, operationalisieren und verwalten.

Ihrer Anwendung herausragende Merkmale hinzufügen



Entwickler-Services

Visual Studio Team Services

Visual Studio Team Services bietet ein Set an Cloud-getriebenen Tools zur Zusammenarbeit, die mit Ihrer vorhandenen IDE bzw. Ihrem vorhandenen Editor funktionieren. Dadurch kann Ihr Team effizient an Software-Projekten in allen Formen und Größe arbeiten, in jeder Sprache und auf jeder Plattform. Diese Tools decken den gesamten Software-Lebenszyklus ab – vom Zeitpunkt des Hinzufügens einer Aufgabe zu den Backlog/Plänen über das Coding für die Aufgabe bis hin Auslieferung in einem Release.

- **Tools für agile Teams:** Erfassen, priorisieren und verfolgen Sie die Arbeiten mit Backlogs und anpassbaren Kanban-Boards. Arbeitsaufgaben sind direkt mit dem Code verknüpft, um Transparenz zu gewährleisten, und können verwendet werden, um reichhaltige Dashboards für ein einfaches Reporting zu erstellen.
- **Versionskontrolle:** Speichern und arbeiten Sie gemeinsam an Code mit unbegrenzten privaten Repositories und Nuget Package-Management-Service. Verwenden Sie Git zur verteilten Versionskontrolle, um die Zusammenarbeit zu maximieren, oder verwenden Sie Team Foundation Version Control (TFVC) zur zentralisierten Versionskontrolle.
- **Continuous Integration und Cross-Plattform-Builds:** Fangen Sie Qualitätsprobleme frühzeitig mit Continuous Integration (CI) -Builds ab, die Ihre Anwendung nach jeder Code-Änderung automatisch kompilieren und testen. Nutzen Sie Continuous Delivery, um Anwendungen oder Websites, die die Tests bestehen, automatisch bereitzustellen. Richten Sie Release-Umgebungen und Richtlinien ein, um Release-Pipelines zu verwalten.
- **IDE-Integration:** Verwenden Sie Ihre bevorzugte Sprache und Ihr bevorzugtes Entwicklungs-Tool. Die Versionskontrolle unterstützt jede Sprache sowie jeden Git-Client (einschließlich Xcode). Java-Teams können auf Code und Work-Items über freie Plug-ins für Eclipse, Android Studio und IntelliJ zugreifen – und Continuous Integration-Builds basierend auf Konfigurationsdateien von Ant oder Maven ausführen.

Dev-Test Labs

Dev-Test Labs berücksichtigt schnelle, leichte und schlanke Dev/Test-Umgebungen in der Cloud. Entwicklungs- und Testumgebungen können Sie schnell provisionieren, indem Sie virtuelle Maschinen über wiederverwendbare Vorlagen oder Azure Marketplace mit wenigen Klicks erstellen. Das gestattet es Benutzern, Azure Marketplace-Images als Basis für virtuelle Maschine im Labor zu nutzen, zusätzlich zu ihren eigenen maßgeschneiderten Images (VHDs), die in das Lab hochgeladen sind. Mit Quotas und Richtlinien minimieren Sie Verluste, und Sie das Herunterfahren automatisieren, um Kosten zu sparen. Alle Vorteile resultieren auf der Verwendung der Cloud für Ihre Entwicklungs- und Testumgebungen – sowohl für Windows- als auch für Linux-Workloads.

HockeyApp

HockeyApp vereinfacht die Verteilung, das Testen sowie das Einholen von Feedback bei Ihren mobilen Client-Anwendungen. Unterstützt werden iOS, Android oder Windows ebenso wie nativer Code. Xamarin-, Cordova- und Unity-Unterstützung decken die Open Source HockeyApp-SDKs ab.

HockeyApp bietet Webhooks und ein API für die Integration in Ihre vorhandenen ALM-Tools. Das erlaubt es Ihnen, eine Mobile DevOps-Praxis zu übernehmen, ohne dafür Ihre bestehenden Workflows zu ändern. HockeyApp unterstützt die Work-Item-Erstellung basierend auf Anwendungs-Feedback und -Abstürzen, sodass Sie Ihr Backlog an einer einzigen Stelle zentral verwalten können.

Application Insights

Application Insights versieht Sie mit reichhaltigen Möglichkeiten zur Leistungsüberwachung von Webanwendungen und -services sowie umfangreichen Diagnoseinformationen, um auf Probleme schnell reagieren zu können. Unterstützt werden Public, Private und Hybrid Clouds sowie on-premises Bereitstellungen – und nahezu jede Webentwicklungssprache. Application Insights kann Sie mit den Erkennungs-, Selektierungs- und Diagnosefähigkeiten versehen, die Sie benötigen, um Ihren Webanwendungen und -services jederzeit zu vertrauen.

Ihrer Anwendung herausragende Merkmale hinzufügen



Identität

Azure Active Directory (AD) bietet Unternehmen eine Identitätsverwaltung für Cloud-Anwendungen auf Enterprise-Niveau. Die Azure AD-Integration versieht Ihr Betriebssystem mit einem optimierten Anmeldeerlebnis und hilft Ihren Anwendungen bei der Konformität zu IT-Richtlinien.

Unterstützung von Azure AD als eine Möglichkeit zur Anmeldung bei Ihrer Anwendung

- **Verwendung von Azure AD zur Anmeldung bei Ihrer Anwendung:** Ihre Benutzer müssen sich keinen weiteren Namen und kein weiteres Kennwort merken, während müssen ein Passwort weniger speichern und schützen müssen. Azure AD ist die treibende Kraft hinter Anmeldungen bei einigen der weltweit beliebtesten Cloud-Anwendungen, einschließlich Office 365 und Microsoft Azure. Erfahren Sie mehr über das [Hinzufügen der Unterstützung für die Azure AD-Anmeldung](#).
- **Vereinfachen Sie die Anmeldung für Ihre Anwendungen:** Während der Registrierung für Ihre Anwendung kann Azure AD wesentliche Informationen über einen Benutzer liefern, sodass Sie imstande sind, das Registrierungsformular vorauszufüllen oder dieses vollständig abzuschaffen. Benutzer können sich für Ihre Anwendung mit ihrem Azure AD-Konto über ein vertrautes Genehmigungsverfahren registrieren, ähnlich denen in Social Media- und mobilen Anwendungen. Erfahren Sie mehr über die [Registrierung Ihrer Anwendung für die Azure AD-Kontenanmeldung](#).

Suche nach Benutzern, Verwaltung der Benutzerprovisionierung und Kontrolle des Zugriffs auf Ihre Anwendung

- **Suche nach Benutzern im Verzeichnis:** Verwenden Sie das Graph API, um Benutzern dabei zu helfen, nach anderen Personen in ihrem Unternehmen zu suchen, wenn sie diese einladen oder ihnen Zugriff gewähren möchten – sodass es für diese Personen nicht erforderlich ist, E-Mail-Adressen einzugeben. Erfahren Sie mehr über das [Graph API](#).

- **Wiederverwendung vorhandener Azure AD-Gruppen und Verteilerlisten:** Azure AD enthält die Gruppen, die Ihr Kunde für die E-Mail-Verteilung und zur Zugriffsverwaltung bereits verwendet. Durch die Wiederverwendung dieser Gruppen braucht Ihr Kunde keine separaten Gruppen in Ihrer Anwendung zu erstellen und zu verwalten.
- **Azure AD nutzen zur Kontrolle, wer Zugriff auf Ihre Anwendung besitzt:** Administratoren können bestimmten Benutzern und Gruppen den Zugriff auf Anwendungen zuweisen. Lesen Sie diese Liste aus und nutzen Sie diese innerhalb Ihrer Anwendung zur Steuerung der Provisionierung und De-Provisionierung von Ressourcen und Zugriffsmöglichkeiten.

Erweiterte Sicherheitsfunktionen

- **Multifaktor-Authentifizierung:** Azure AD bietet native Multifaktor-Authentifizierung. IT-Administratoren können die Multifaktor-Authentifizierung zum Zugriff auf Ihre Anwendung verlangen, und so brauchen Sie für diese Funktion nicht selbst Code zu schreiben. Erfahren Sie mehr über die [Multifaktor-Authentifizierung](#).
- **Erkennung von Anmeldeanomalien:** Azure AD verarbeitet mehr als eine Milliarde Anmeldungen pro Tag und nutzt dabei Machine-Learning-Algorithmen, um verdächtige Aktivitäten zu erkennen und IT-Administratoren über mögliche Probleme zu informieren. Durch die Unterstützung der Azure AD-Anmeldung werden auch Ihrer Anwendung die Vorteile dieser Schutzmechanismen zuteil. Erfahren Sie mehr über die [Zugriffs- und Nutzungsberichtsanzeige von Azure AD](#).

Microsoft Graph

Microsoft Graph veröffentlicht mehrere APIs der Microsoft Cloud Services über einen einzelnen REST API-Endpunkt (<https://graph.microsoft.com>). Microsoft Graph gestattet:

- Einen einheitlichen API-Endpunkt für den Zugriff auf die von mehreren Microsoft Cloud Services stammenden, aggregierten Daten in einer einzigen Antwort.
- Nahtlose Navigation zwischen Entities und den Beziehungen zwischen ihnen.
- Zugriff auf Erkenntnisse und Einblicke, die von der Microsoft Cloud kommen.

Ihrer Anwendung herausragende Merkmale hinzufügen



Integration

Azure bietet eine große Vielzahl an Integrationsdiensten, die es Ihnen erlauben, Integrationslösungen auf die Cloud auszuweiten.

BizTalk Services

BizTalk Services bieten serienmäßig die Cloud-zum-lokalen-Rechenzentrum- und Branchenanwendungs-Integration für SAP, Oracle EBS, SQL Server und PeopleSoft. Sie können sich mit beliebigen HTTP-, FTP-, SFTP- oder REST-Datenquellen verbinden. Nachrichten lassen sich weiterleiten durch Verwendung verschiedener Azure-Artefakte wie Service Bus-Queues, Topics, SQL-Datenbank und Blob-Speicher.

Hybride Verbindungen

Das hybride Verbindungs-Feature der BizTalk Services gestattet es Ihnen, Azure Websites oder Azure Mobile Services mit jeder on-premises TCP- oder HTTP-Ressource zu verbinden – wie zum Beispiel Microsoft SQL Server, MySQL oder ein beliebiger Webservice. Dazu werden lediglich ein paar Konfigurationsänderungen benötigt. Die Verwendung von benutzerdefiniertem Code ist nicht erforderlich.

Service Bus

Azure Service Bus ist ein generisches, Cloud-basiertes Messaging-System für Verbindungen von so gut wie allem – Anwendungen, Services und Geräte – wo auch immer sich diese befinden. Sie können auf Azure laufende Anwendungen, on-premises Systeme oder beides verbinden. Sie können Service Bus sogar dazu verwenden, Haushaltsgeräte, Sensoren und andere Geräte wie Tablets oder Telefone mit einer zentralen Anwendung oder untereinander zu verbinden.

Queues

Azure unterstützt zwei Arten von Warteschlangenmechanismen: Azure Queues und Service Bus Queues.

- Azure Queues sind Teil der Azure-Speicherinfrastruktur und weisen eine einfache, REST-basierte Get/Put/Peek-Schnittstelle auf. Sie bieten ein zuverlässiges, persistentes Messaging innerhalb und zwischen Services.
- Service Bus Queues sind Bestandteil einer breiteren Azure-Messaging-Infrastruktur, die das Queuing ebenso unterstützt wie Publish/Subscribe, Webservice-Remoting und Integrationsmuster.

Mehr dazu erfahren Sie bei [Überblick über das Service Bus-Messaging](#).

Azure Queues wurden zunächst als dedizierter Warteschlangen-Speichermechanismus eingeführt, der auf den Azure-Speicher-Services basiert. Service Bus Queues basieren auf der breiter gefassten „Brokered Messaging“-Infrastruktur. Diese wurde dafür geschaffen, Anwendungen oder Anwendungskomponenten zu integrieren, die sich über mehrere Kommunikationsprotokolle, Datenkontrakte, vertraute Domänen und/oder Netzwerkkumgebungen erstrecken können.

Ihrer Anwendung herausragende Merkmale hinzufügen



Media Services & CDN

Media Services

Azure Media Services kombiniert leistungsstarke und hochgradig skalierbare Cloud-basierte Codierungs-, Verschlüsselungs- und Streaming-Komponenten. Auf diese Weise können Kunden Premium-Video-Inhalte an ein größeres Publikum auf die heutzutage beliebtesten digitalen Geräte wie Tablets und Mobiltelefone liefern.

- **Encoding:** Vom der einfachen Weblieferung für HTML5 bis zur komplexen Medien-Entscheidungslogik, die anspruchsvolle Studio-Workflows löst, erlaubt es Azure Media Services Ihnen, Ihre eigenen Encodierungs-Workflows zu definieren.
- **Indizierung:** Ermöglichen Sie die tiefgehende Suche in Ihren Medieninhalten. Unter Verwendung von Deep Neural Net (DNN) -basierter Technologie zur Spracherkennung konvertiert der Media Indexer digitale Audioinformationen in natürliche Sprache und entnimmt Ihren Medien automatisch sinnvolle Metadaten.
- **Content-Schutz:** Azure Media Services ist die einzige Cloud-Medienlösung, die eine spontane Verschlüsselung sowohl für Video-On-Demand (VOD) als auch Live-Streams bietet.
- **Azure Media Player:** Wählt automatisch das beste Format für einen Browser oder ein Gerät. Zur Wiedergabe adaptiver Streaming-Inhalte in verschiedenen Formaten werden die dynamischen Packaging-Fähigkeiten der Azure Media Services genutzt. Für Entwickler gibt es eine einfache, einheitliche Schnittstelle zum Zugriff auf APIs.

Gängige Szenarien

- **Geschäftliche Videoinhalte in Ihrem Unternehmen bereitstellen**
Die Fähigkeiten der Plattform, kombiniert mit Partnerlösungen, machen es einfach, Video in Ihre Anwendungen und Ihr Unternehmen zu integrieren – für Trainingszwecke, zur Unternehmenskommunikation oder in Meetings. Media Services bietet eine skalierbare, jederzeit verfügbare und sichere Bereitstellung von Videoinhalten sowohl an Mitarbeiter als auch externe Kunden via Azure.
- **Übertragungen streamen und OTT-Video**
Schaffen Sie sich ein Online-Publikum und vergrößern Sie Ihre Reichweite durch die Verteilung Ihrer Inhalte an die Telefone, Tablets und alle digitalen Geräte Ihrer Zuschauer. Darüber hinaus können Sie von aktuellen Ereignissen oder Nischenmärkten profitieren, indem Sie schnell neue Kanäle übertragen und Over-The-Top (OTT) -Videoinhalte in mehreren Formaten encodieren.
- **Regierungsinhalte bereitstellen**
Staatliche Stellen können sicheres Video-Streaming auf mobile Geräte vornehmen, indem Sie Azure Media Services und Azure Government verwenden, eine CJIS (Criminal Justice Information Services) -konforme und geschützte Plattform.
- **Inhalte näher an Ihre Benutzer heranbringen**
Azure Content Delivery Network lässt Sie Inhalte mit hoher Bandbreite über ein robustes Netzwerk von globalen Datenzentren an Benutzer auf der ganzen Welt mit niedriger Latenz und hoher Verfügbarkeit liefern. Es sendet Audio, Video, Anwendungen, Bilder und andere Dateien an Benutzer von den jeweils nächstgelegenen Servern.

Fazit

Ähnlich wie Entwickler, kommen auch Anwendungen in unterschiedlichsten Ausführungen. Einst aufgrund Umfang und Komplexität – oder, weil man sie sich schlicht und einfach nicht vorstellen konnte – nicht realisierbare Anwendungen sind jetzt mit der Cloud zur Realität geworden. In diesem Handbuch haben wir die Azure Anwendungs-Plattform-Services beleuchtet und dabei insbesondere die Azure-Unterstützung für Platform-as-a-Service (PaaS) besonders hervorgehoben.

Warum sollten Sie PaaS als Ihre standardmäßige Wahl betrachten? Lassen Sie uns dazu ein letztes Mal auf eine Analogie zurückgreifen.

Telefone bieten enorme soziale und geschäftliche Vorteile. Aber die Verlegung und Pflege Millionen Kilometer von Kabeln zur Unterstützung einer allgegenwärtigen Infrastruktur ist teuer. Heute wächst die Telefonnutzung am schnellsten in Afrika. Aber folgt man dort dem Weg von Europa, Amerika und anderen Regionen, die auf das Festnetz setzten? Das ist nicht der Fall: Tatsächlich erlebten dort Telefonkabelinstallationen im Jahr 2009 einen Höhepunkt, sodass nur vier Prozent der Bevölkerung Zugang zu einem Festnetzanschluss haben. Stattdessen überspringt Afrika die Telefonkabel-Technik und wendet sich direkt der Mobiltelefon-Technik zu, die viel günstiger zu installieren und zu warten ist.

In vielerlei Hinsicht ist PaaS wie das Mobiltelefon. Es bietet eine höhere Funktionalität, niedrigere Anfangsinvestitionen und eine höhere Entwicklerproduktivität, als es bei IaaS der Fall ist. Zudem entfällt ein Großteil der ursprünglichen Infrastrukturanforderungen, während laufende Wartungskosten verringert werden.

Azure bietet Entwicklern eine umfassende PaaS-Anwendungsplattform für die Erstellung, Bereitstellung und Verwaltung aller Arten von Anwendungen – von der einfachsten Website bis hin zu den komplexesten Geschäftslösungen.

**Willkommen zur nächsten Entwickler-Welle.
Willkommen bei Azure.**

Empfohlene nächste Schritte

- Seien Sie unser Gast bei einem bis zu einstündigen [Azure App Service-Erlebnis](#) – ohne Abonnement, ohne Kosten und ohne weitere Bindung.
- Entdecken Sie die große Auswahl an [kostenlos für Ihren Einstieg zur Verfügung stehende Optionen](#): Hosten Sie bis zu zehn kostenlose Web- und mobile Anwendungen auf Azure App Service, versenden Sie bis zu eine Million Benachrichtigungen mit Notification Hubs, oder führen Sie Machine Learning-Experimente durch.
- Erstellen Sie ein Azure-Konto und [starten Sie kostenlos](#) mit einer Azure-Gutschrift in Höhe von \$ 200.