

Proof-of-Concept-Leitfaden für Azure-Entwickler



Proof-of-Concept-Leitfaden für Azure-Entwickler

03 /

Einführung

- 4 Was ist ein Proof of Concept?
- 5 Über diesen Leitfaden
- 5 Was ist Microsoft Azure?

06 /

Kapitel 1: Proof-of-Concept-Leitfaden

- 6 Schritt 1: Ziele und Erfolgskriterien definieren
- 6 Schritt 2: Zeitrahmen und Budget festlegen
- 7 Schritt 3: Umfang des Proof-of-Concept-Projekts festlegen
- 8 Schritt 4: Allgemeine Architektur erstellen
- 9 Schritt 5: Team zusammenstellen
- 9 Schritt 6: Implementieren und Testen
- 9 Schritt 7: Nach Abschluss Ihres Proof of Concept

11 /

Kapitel 2: Beispielprojekt – Implementieren einer Web-App mit Azure Static Web Apps

- 13 Azure Static Web Apps kontra traditionelle Webserver
- 13 Hauptmerkmale von Azure Static Web Apps
- 13 Anwendungsfälle für Azure Static Web Apps
- 14 Praktische Umsetzung
- 21 Weiterführende Informationen

22 /

Kapitel 3: Beispielprojekt – Entwicklung eines intelligenten Chatbots

- 22 Hauptmerkmale von Chatbots
- 24 Anwendungsfälle für Chatbots
- 25 Praktische Umsetzung

36 /

Kapitel 4: Übersicht über Azure für Entwickler

- 36 Erste Schritte mit Microsoft Azure
- 36 Die Vorteile von Azure
- 37 Azure-Abonnement
- 37 Was umfasst das kostenfreie Azure-Konto?

50 /

Kapitel 5: Weiterführende Lernmaterialien und Ressourcen

- 50 Grundlagen von Azure erlernen
- 50 Die für das Entwerfen eines Azure-Proof-of-Concept-Projekts nötigen Tools
- 51 Weitere nützliche Ressourcen

52 /

Schlussbemerkung

© 2020 Microsoft Corporation. Alle Rechte vorbehalten.

Dieses Dokument wird ohne Mängelgewähr zur Verfügung gestellt. Die hierin dargelegten Informationen und Ansichten, einschließlich URLs und anderer Verweise auf Websites, können ohne vorherige Ankündigung geändert werden. Sie tragen das Risiko der Nutzung. Mit diesem Dokument erhalten Sie keinerlei Rechte an geistigem Eigentum eines Microsoft-Produkts. Dieses Dokument kann zu internen Referenzzwecken vervielfältigt und verwendet werden.

Einführung

Aufgrund des Vormarschs neuer Technologien führen viele Unternehmen Proof-of-Concept-Projekte durch, um neue Funktionen zu evaluieren und vorgeschlagene Konzepte zu bewerten. Ein Proof of Concept stellt einen wichtigen ersten Schritt zur Förderung von Geschäftsinnovationen dar.

Laut einer kürzlich von Packt Publishing durchgeführten Umfrage (siehe Abbildung 1: So werden Tools/Plattformen von Entwicklern bewertet und eingeführt) ist ein Proof of Concept die beliebteste Methode für Entwickler zur Bewertung und Einführung neuer Tools und Plattformen.

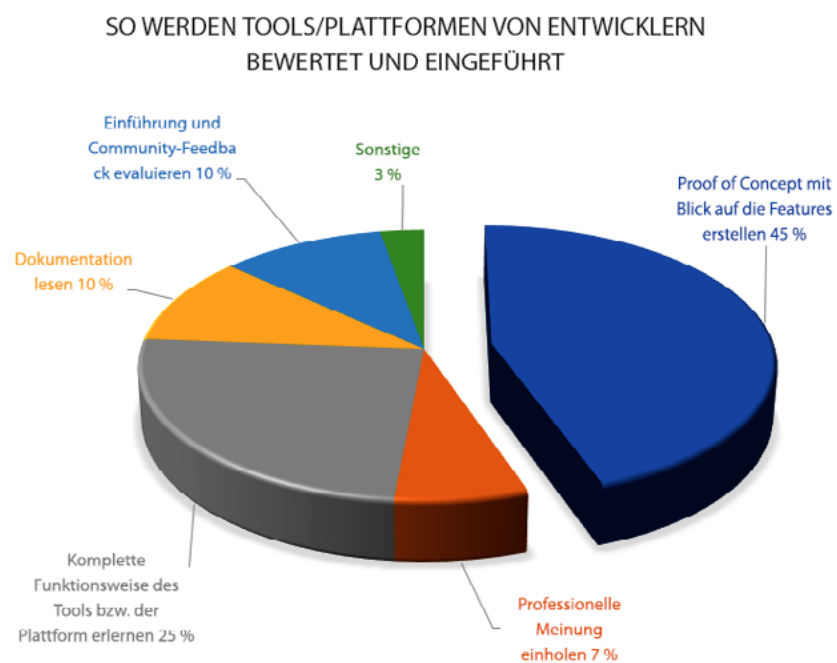


Abbildung 1: So werden Tools/Plattformen von Entwicklern bewertet und eingeführt

Was ist ein Proof of Concept?

Bei einem Proof of Concept bzw. Machbarkeitsnachweis handelt es sich um eine klar umrissene, zeitlich begrenzte Maßnahme mit spezifischen, messbaren Zielen und Erfolgskennzahlen. Um sinnvolle Ergebnisse zu erhalten, sollte er zu einem gewissen Grad auf der geschäftlichen Praxis basieren.

Die Vorteile von Proof-of-Concept-Projekten

Ein Proof-of-Concept-Projekt stellt ein wertvolles Instrument dar, mit dem sich bewerten lässt, ob eine potenzielle Technologie oder ein Konzept die Anforderungen einer Geschäftslösung erfüllt. Damit lassen sich mögliche technische und logistische Probleme identifizieren, bevor der Dienst in einem regulären Projekt implementiert wird. Ferner bietet es frühzeitige Insights in die Technologie und verringert Risiken, da wichtige Entscheidungen schon in der Anfangsphase des Entwicklungsprozesses getroffen werden können.

Proof-of-Concept-Projekte bieten mehrere Vorteile:

Experimentieren mit neuen Technologien

Angesichts der Weiterentwicklung von Technologien können Unternehmen mithilfe von Proof-of-Concept-Projekten neue Technologien evaluieren, verstehen und damit experimentieren, mit dem Ziel, sie eventuell in kommenden Projekten einzusetzen. Im Rahmen eines klar umrissenen und zeitlich festgelegten Ablaufs kann das Entwicklungsteam schnell mit neuer Technologie arbeiten, ohne dass die regulären Großprojekte des Unternehmens gefährdet wären. Das Beste daran ist, dass das Ergebnis des Proof-of-Concept-Projekts in Zukunft in einem regulären Projekt aufgegriffen werden kann. Daher fördern solche Projekte Innovationen.

Minimierung von Risiken

Vor Beginn eines risikoreichen, groß angelegten und potenziell teuren Projekts ist es empfehlenswert, Risiken und Kosten zu minimieren. Dazu sollte eine zügige Prüfung der als riskant eingestuften Teile des Projekts erfolgen. Beispielsweise könnte ein Projektteam eine bestimmte, in einem Großprojekt eingesetzte Technologie identifizieren und isolieren und sie aufgrund der geringen Erfahrung des Entwicklungsteams mit dieser Technologie als riskant einstufen. Daraufhin kann das Projektteam eine schnelle Bewertung der genannten Technologie vornehmen, indem es ein kleines Teilprojekt mit einem festen Budget und Zeitplan erstellt. Risiken werden reduziert, da Proof-of-Concept-Projekte in der Regel in einer kontrollierten Sandbox-Umgebung durchgeführt werden.

Über diesen Leitfaden

Der Zweck dieses Leitfadens besteht darin, den Leser bei der Planung eines erfolgreichen Proof-of-Concept-Projekts zu unterstützen. Außerdem wird erläutert, wie Sie auf der Cloud-Plattform Microsoft Azure mit der Entwicklung beginnen können. Dieser Leitfaden richtet sich an Entwickler und Architekten, die gerade erst mit Azure loslegen.

Die in „Kapitel 1: Proof-of-Concept-Leitfaden“ dargelegte Methodik bildet die zentrale Grundlage für die Planung und Durchführung eines erfolgreichen Proof-of-Concept-Projekts.

In „Kapitel 2: Beispielprojekt – Implementieren einer Web-App mit Azure Static Web Apps“ und „Kapitel 3: Beispielprojekt – Entwicklung eines intelligenten Chatbots“ werden praktische Projekte vorgestellt, die Sie als Inspiration für Ihre eigenen Proof-of-Concept-Projekte nutzen können.

In „Kapitel 4: Übersicht über Azure für Entwickler“ stellen wir Ihnen die Azure-Plattform vor. Neueinsteiger erfahren, wie sie ein kostenfreies Azure-Konto anlegen, um mit Proofs of Concepts auf Azure loszulegen. Außerdem erfahren Sie, wie Sie das geeignete Cloudmodell und die passenden Dienste auswählen, um mit der Entwicklung Ihres Projekts für Azure zu beginnen.

In „Kapitel 5: Weiterführende Lernmaterialien und Ressourcen“ finden Sie zudem einige nützliche Lernmaterialien.

Was ist Microsoft Azure?

Dieser Leitfaden befasst sich schwerpunktmäßig mit Microsoft Azure. Azure ist eine Cloudplattform, mit der Sie Ihre vorhandenen Anwendungen mit minimalen Änderungen hosten und neue Cloudanwendungen entwickeln können. Es gibt eine Vielzahl von einsatzbereiten Azure-Diensten, die Sie in Ihre Anwendungen integrieren können. So profitieren Sie umgehend von den neuen Cloudfunktionen, ohne derartige Dienste selbst entwickeln zu müssen.

Wenn Sie Ihre Anwendungen auf Azure hosten, können Sie ein Minimum Viable Product entwickeln und danach Ihre Anwendung gemäß wachsender Kundennachfrage skalieren. Azure bietet die für hoch verfügbare Anwendungen erforderliche Zuverlässigkeit, sogar bei einem Failover zwischen verschiedenen Regionen. Mit dem Azure-Portal können Sie alle Ihre Azure-Dienste ganz einfach verwalten. Auch die programmatische Verwaltung Ihrer Dienste mithilfe von dienstspezifischen APIs und Vorlagen ist möglich.

In „Kapitel 4: Übersicht über Azure für Entwickler“ erfahren Sie mehr über Azure.

Doch zunächst beginnen wir mit dem Proof-of-Concept-Leitfaden.



Kapitel 1: Proof-of-Concept-Leitfaden

Schritt 1: Ziele und Erfolgskriterien definieren

Die meisten Proof-of-Concept-Projekte sind ergebnisorientiert. Legen Sie das Ziel fest, das Sie mit Ihrem Proof-of-Concept-Projekt erreichen möchten, und bestimmen Sie anhand von Erfolgskriterien, wann ein Projekt als erfolgreich zu betrachten ist.

Es soll nachgewiesen werden, ob ein Konzept funktioniert oder nicht, ohne eine enorme Menge an Zeit und Ressourcen zu investieren. Wenn das Proof-of-Concept-Projekt erfolgreich ist und die erwarteten Ergebnisse eintreten, kann das Projektteam grünes Licht für die nächsten Schritte geben. Sollte das Experiment jedoch scheitern, so geschieht das innerhalb eines festgelegten Budgets, Zeitrahmens, Umfangs und mit festgelegten Ressourcen, was im Großen und Ganzen nur minimale Kosten verursacht. Der Vorteil eines Proof-of-Concept-Projekts liegt darin, dass das Projektteam schnell entscheiden kann, ob das Konzept weiterverfolgt, aufgegeben oder eine Alternative ausgewählt werden soll. Berücksichtigen Sie beim Entwerfen Ihres Projekts (siehe Schritt 3: Umfang des Proof-of-Concept-Projekts festlegen), dass das Ziel Ihres Proof of Concept darin besteht, schnell erfolgreich zu sein (bzw. schnell zu scheitern), damit Folgeentscheidungen zeitnah getroffen werden können.

Schritt 2: Zeitrahmen und Budget festlegen

Sobald die Ziele und Erfolgskriterien des Proof-of-Concept-Projekts definiert sind, ist der zeitliche Rahmen und das Budget festzulegen.

Unserer Erfahrung nach erzielen Proof-of-Concept-Projekte das beste Ergebnis, wenn der Zeitplan auf zwei bis vier Wochen beschränkt ist. Das bietet genügend Zeit für den Abschluss der Arbeiten, ohne sich zu viele Fälle und komplexe Testmatrizen aufzubürden.

Einige Tipps:

- Schätzen Sie den erforderlichen Zeitaufwand für das Erledigen der Aufgaben des Proof of Concept realistisch ein.
- Wenn Sie der Ansicht sind, dass für das Proof-of-Concept-Projekt mehr als vier Wochen nötig sind, sollten Sie erwägen, dessen Umfang zu reduzieren, um sich auf die Ziele mit der höchsten Priorität zu konzentrieren.
- Lassen Sie den Zeitrahmen und das Budget vor dem Start von allen wichtigen Personen und Geldgebern absegnen.

Nach dem Festlegen des Zeitplans und des Budgets muss der Umfang des Proof of Concept definiert werden.

Schritt 3: Umfang des Proof-of-Concept-Projekts festlegen

Es ist wichtig, den Umfang Ihres Proof-of-Concept-Projekts zu definieren, bevor Sie mit der Implementierung beginnen, um eine schleichende Umfangserweiterung zu verhindern. Stakeholder setzen eher dann angemessene Erwartungen an den Proof of Concept, wenn im Voraus klar ist, dass die Ressourcen nicht auf unbegrenzte Zeit zur Verfügung stehen.

Tip: Wenn Sie den Umfang vor Beginn festlegen, vermeiden Sie es, dass sich Proof-of-Concept-Projekte in Produktionsprojekte verwandeln. Stellen Sie im Laufe des Projekts sicher, dass alle Beteiligten den Umfang einhalten.

Die folgenden wichtigen Fragen helfen Ihnen, den Umfang Ihres Proof-of-Concept-Projekts zu ermitteln:

- Was möchten Sie mit Ihrem Proof of Concept erreichen bzw. was möchten Sie daraus lernen?
- Wie lauten die Erfolgskriterien?
- Welche Workloads oder Szenarien werden behandelt?

Tip: Damit Ihr Proof-of-Concept-Projekt rechtzeitig umrissen und abgeschlossen wird, sollten die Workloads und Szenarien möglichst klein ausfallen.

- Welche Ressourcen werden benötigt?
- Welche Benutzer und Teams prüfen das Ergebnis?
- Wie viel Zeit ist für das Projekt veranschlagt?

Tip: Wählen Sie einen Zeitrahmen, der sich nach dem geplanten Umfang richtet, z. B. zwei oder vier Wochen.

- Was geschieht mit den während des Projekts zugewiesenen Ressourcen nach Abschluss des Proof of Concept? Planen Sie, diese Ressourcen zu streichen?

Nachdem der Umfang definiert ist, muss eine allgemeine Architektur für den Proof of Concept erstellt werden.

Schritt 4: Allgemeine Architektur erstellen

Zur Erinnerung: Das Ziel eines Proof of Concept ist es, ein Konzept auf dessen Tragfähigkeit hin zu überprüfen. Damit das Ziel Ihres Projekts innerhalb des festgelegten Zeitrahmens erreichbar ist, müssen Sie bei der Erstellung der allgemeinen Architektur entscheiden, welche wesentlichen Komponenten Teil des Proof of Concept sind und welche unwesentlichen Komponenten davon auszuschließen sind.

Als Faustregel gilt: Je mehr Komponenten Sie in Ihren Proof of Concept einbeziehen, desto komplizierter wird das Projekt, und desto länger dauert es, die Erfolgskriterien zu erfüllen.

Als Teil Ihrer allgemeinen Architektur müssen Sie auch das geeignete Cloudmodell für das Hosten Ihrer Proof-of-Concept-Anwendung auswählen. Mehr dazu unter „How to choose the appropriate cloud model for your Azure proof of concept project“ in Kapitel 4: Übersicht über Azure für Entwickler.

Es empfiehlt sich, Ihre Proof-of-Concept-Projekte immer in einem von der Produktion getrennten Azure-Abonnement zu speichern. Idealerweise sollten Sie ein Dev/Test-Abonnement verwenden, um die Kosten niedrig zu halten. Mehr über Azure-Abonnements erfahren Sie im Abschnitt „Getting started with Microsoft Azure“ in Kapitel 4: Übersicht über Azure für Entwickler.

Tipp: Perfektion ist nicht alles. Zwar ist es verlockend, einen perfekten Proof of Concept zu entwerfen, der das spätere Endprodukt widerspiegelt. Für ein Proof-of-Concept-Projekt wäre das jedoch kontraproduktiv. Je stärker Sie nach Perfektion streben, desto mehr Zeit und Aufwand müssen Sie vom Start bis zum Ende investieren. Das ist aber nicht erstrebenswert. Wie Sie sich erinnern, besteht der Zweck des Proof of Concept darin, ein bestimmtes, klar umrissenes Konzept zügig zu bewerten. Es dient dazu, zeitnah die richtigen Entscheidungen treffen zu können. Daher sollte der Schwerpunkt stets auf der Auswahl der kleinsten wesentlichen Abhängigkeiten und der damit verbundenen Workloads liegen. Diese müssen bestimmte messbare Ziele erreichen können, um so einen raschen Erfolg zu ermöglichen.

Nachdem der Architekturplan vorliegt, sollten Sie Ihr Team für das Proof-of-Concept-Projekt zusammenstellen.

Schritt 5: Team zusammenstellen

Für ein kleines, einmaliges, einfaches, unkritisches, experimentelles Proof-of-Concept-Projekt ist ein Ein-Personen-Team ausreichend. Für die meisten typischen Proof-of-Concept-Projekte, bei denen die Ergebnisse von großer Bedeutung sind und die Entscheidungen im Rahmen eines größeren Projekts beeinflussen könnten, sollten Sie jedoch die zur Unterstützung erforderlichen Teammitglieder und das nötige Commitment identifizieren. Das von Ihnen zusammengestellte Team muss den Umfang des Projekts widerspiegeln.

Schritt 6: Implementieren und Testen

Wenn das Ziel, der Zeitrahmen, das Budget und der Umfang definiert sind und Ihr Team zusammengestellt ist, können Sie mit der Umsetzung Ihres Proof-of-Concept-Projekts basierend auf der allgemeinen Architektur beginnen. Befolgen Sie moderne DevOps-Prozesse mit iterativer Entwicklung und Tests während der gesamten Implementierung, damit die Durchführung möglichst erfolgreich abläuft.

Schritt 7: Nach Abschluss Ihres Proof of Concept

Bewerten Sie nach Abschluss des Proof of Concept, ob die von Ihnen in „Schritt 1: Ziele und Erfolgskriterien definieren“ definierten Erfolgskriterien erfüllt wurden.

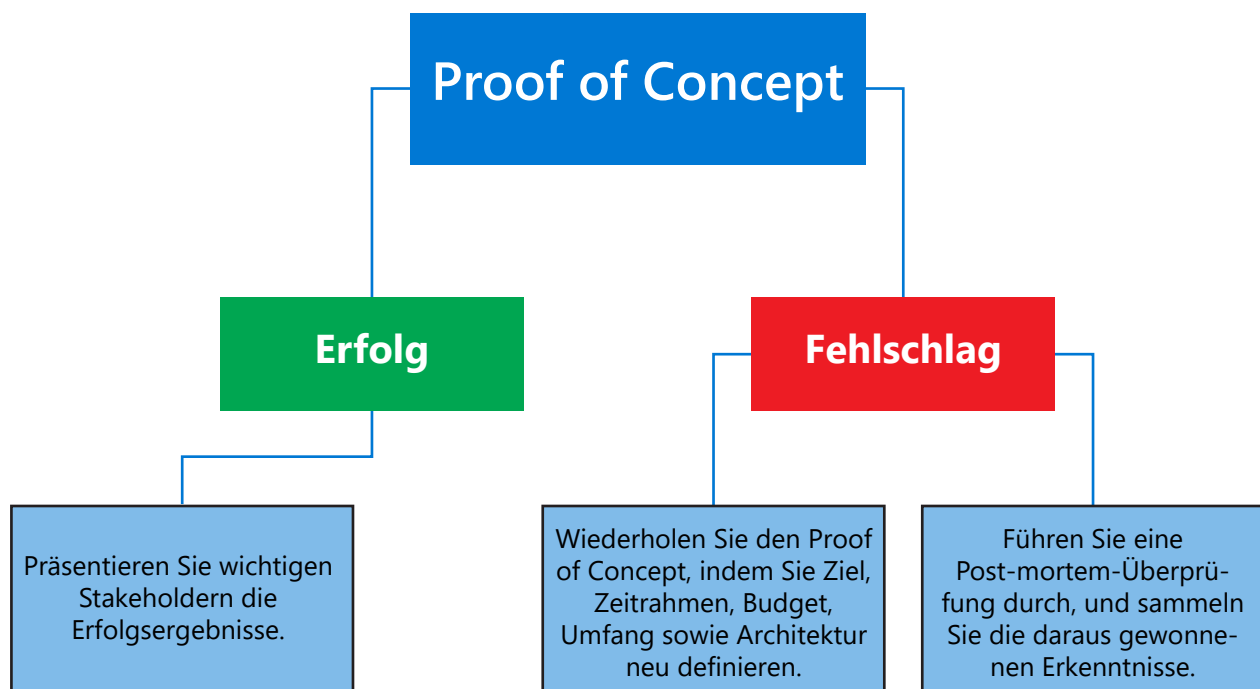


Abbildung 2: Bewerten Ihres Proof of Concept

Bei erfolgreichem Proof of Concept

- Präsentieren Sie Ihre erfolgreichen Ergebnisse wichtigen Stakeholdern.
Tipp: Versuchen Sie, den finanziellen Erfolg auf die Unternehmensrentabilität zu übertragen, wenn Sie wichtige Entscheidungsträger über das erfolgreiche Proof-of-Concept-Projekt informieren. Einige Beispiele:
 - Unser Proof of Concept hat gezeigt, dass sich durch die Optimierung in der neuen Implementierung monatlich X Euro an Azure-Ausgaben einsparen lassen. Wir empfehlen, dieses Konzept in die Produktion zu übertragen.
 - Kunden wünschen sich diese neue Funktionalität. Zudem hat sie sich im Rahmen des Proof of Concept als erfolgreich erwiesen. Wenn wir dieses Konzept in Azure implementieren, steigt der Umsatz voraussichtlich um X % pro Quartal.

Bei fehlgeschlagenem Proof of Concept

- Es gibt zwei Möglichkeiten:
 - Definieren Sie Ziel, Zeitrahmen, Budget, Umfang sowie Architektur neu, und wiederholen Sie das Proof-of-Concept-Projekt.
 - Führen Sie eine Post-mortem-Prüfung durch, um festzustellen, ob während des Proof of Concept Insights oder Erkenntnisse gewonnen wurden.

In diesem Abschnitt haben Sie gelernt, wie Sie das Proof-of-Concept-Projekt planen und durchführen. Im folgenden Abschnitt zeigen wir zwei praktische Projekte samt Schritt-für-Schritt-Anleitung. Wir hoffen, Sie damit inspirieren zu können, Ihre eigenen Proof-of-Concept-Projekte zu erstellen.



Kapitel 2: Beispielprojekt – Implementieren einer Web-App mit Azure Static Web Apps

Im ersten Beispielprojekt lernen Sie, wie Sie eine Web-App mit Azure Static Web Apps implementieren. Mehr über den Azure App Service erfahren Sie in „Kapitel 4: Eine Übersicht über Azure für Entwickler“. Dieses Projekt behandelt eine der erweiterten Hostingoptionen von App Service: Static Web Apps. Entwickler können mithilfe von Static Web Apps statische Inhalte (wie z. B. HTML, CSS und JavaScript) rendern und zugleich die erforderliche dynamische Logik durch das Entwickeln serverloser APIs mit Azure Functions bereitstellen.

Der Static Web Apps-Workflow (siehe Abbildung 3: Workflow bei Azure Static Web Apps) ähnelt dem tagtäglichen Workflow eines Entwicklers. Static Web Apps bietet eine verwaltete CI/CD-Pipeline (Continuous Integration/Continuous Delivery), die automatisch Full-Stack-Web-Apps aus einem GitHub-Repository erstellt und auf Azure bereitstellt. Möglich ist das dank GitHub Actions (weitere Informationen zu GitHub Actions finden Sie in [dieser Dokumentation](#)).

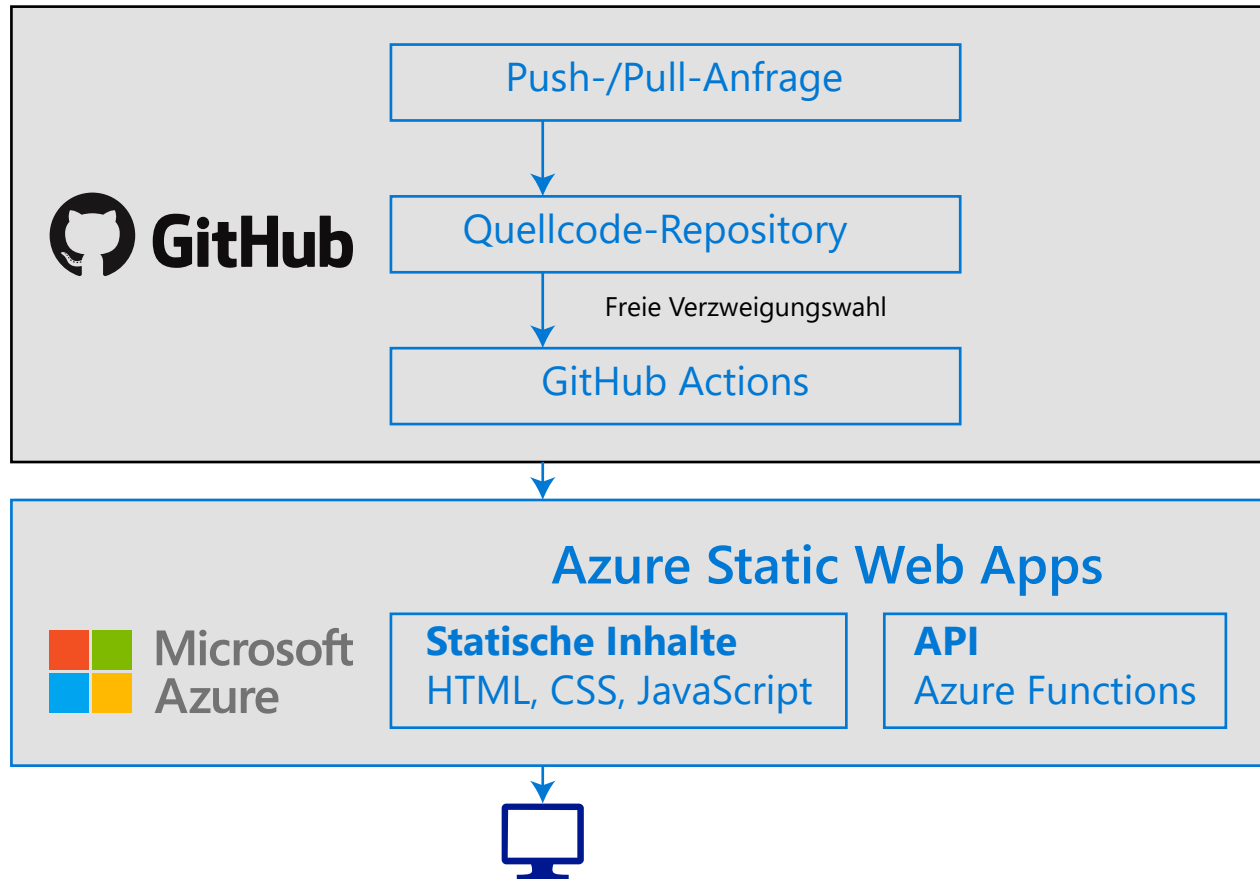


Abbildung 3: Workflow bei Azure Static Web Apps

Beim Erstellen einer Static Web Apps-Ressource richtet Azure einen GitHub Actions-Workflow im Quellcode-Repository der App ein. Dieses Repository überwacht die Verzweigung Ihrer Wahl. Jedes Mal, wenn Sie Commits pushen oder Pull-Anfragen in die überwachte Verzweigung übernehmen, werden Ihre App und deren API automatisch von GitHub Actions entwickelt und auf Azure bereitgestellt.

Azure Static Web Apps werden häufig mithilfe von Bibliotheken und Frameworks wie Angular, React, Svelte oder Vue entwickelt. Diese Apps enthalten HTML-, CSS-, JavaScript- und Image-Ressourcen, aus denen die Anwendung besteht.

Azure Static Web Apps kontra traditionelle Webserver

Bei einem traditionellen Webserver werden die Ressourcen zusammen mit allen erforderlichen API-Endpunkten von einem einzigen Server bereitgestellt.

Bei Static Web Apps werden statische Ressourcen von einem traditionellen Webserver getrennt und stattdessen von weltweit verstreuten Punkten aus bedient. Da der geografische Abstand zwischen Dateien und Anwendern kleiner ist, wird die Bereitstellung von Dateien merklich beschleunigt.

Überdies werden die API-Endpunkte in einer serverlosen Architektur gehostet, wodurch die Notwendigkeit eines vollständigen Back-End-Servers entfällt.

Hauptmerkmale von Azure Static Web Apps

- Web-Hosting für statische Inhalte wie HTML, CSS, JavaScript und Bilder
- Integrierter API-Support durch Azure Functions
- Erstklassige GitHub-Integration, bei der Repository-Änderungen Builds und Bereitstellungen auslösen
- Automatisches Erstellen von Staging-Umgebungen zum Testen von Codeaktualisierungen, bevor diese in die Produktion ausrollen
- Global verteilte statische Inhalte, d. h. Inhalte rücken näher zum Anwender
- Kostenfreie, sich automatisch verlängernde SSL-Zertifikate
- Benutzerdefinierte Domänen für eine markenspezifische Anpassung Ihrer App
- Nahtloses Sicherheitsmodell durch Reverseproxy beim Aufruf von APIs, weshalb keine CORS-Konfiguration erforderlich ist
- Integration von Authentifizierungsanbietern in Azure Active Directory, Facebook, Google, GitHub und Twitter
- Definition und Zuweisung anpassbarer Autorisierungsrollen
- Vollständige Kontrolle über die Inhalte und Routen dank Back-End-Routingregeln

Anwendungsfälle für Azure Static Web Apps

- Entwicklung von Single-Page-Anwendungen mit Frameworks und Bibliotheken wie Angular, React, Svelte oder Vue samt Azure Functions-Back-End
- Veröffentlichung von statischen Websites mit Frameworks wie Gatsby, Hugo und VuePress
- Bereitstellung von Webanwendungen mit Frameworks wie Next.js und Nuxt.js

Praktische Umsetzung



Hinweis: Zum Zeitpunkt der Drucklegung ist Azure Static Web App als Vorschauversion kostenfrei nutzbar. Änderungen sind vorbehalten.

1. Melden Sie sich bei Ihrem GitHub-Konto an, und rufen Sie [diese URL](#) auf, um ein neues Repository zu erstellen.
2. Geben Sie Ihrem Repository einen Namen, z. B. my-first-static-web-app. Klicken Sie danach auf **Create repository from template** (Repository aus Vorlage erstellen):

Create a new repository from angular-basic


The new repository will start with the same files and folders as [staticwebdev/angular-basic](#).


Owner * Repository name *

 / my-first-static-web-app 

Great repository names are short and memorable. Need inspiration? How about [fictional-guide?](#)

Description (optional)

 **Public**
Anyone on the internet can see this repository. You choose who can commit.

 **Private**
You choose who can see and commit to this repository.

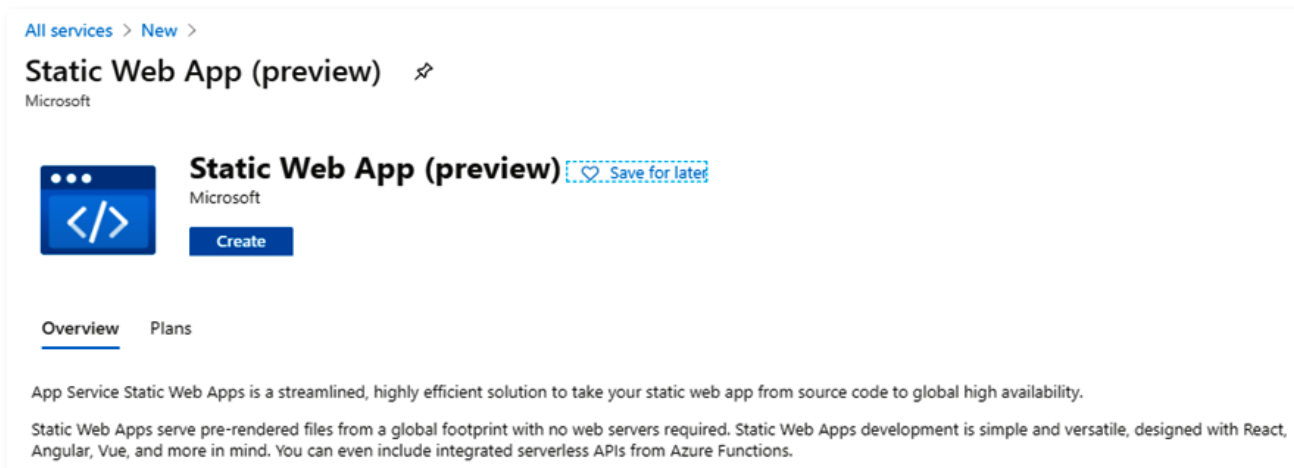
Include all branches
Copy all branches from staticwebdev/angular-basic and not just master.

[Create repository from template](#)

Abbildung 4: Erstellen des GitHub-Repositorys aus der Vorlage

3. Wechseln Sie als Nächstes zum [Azure-Portal](#).
4. Klicken Sie auf **Create a resource** (Ressource erstellen), um zum Azure Marketplace zu gelangen. Geben Sie in das Suchfeld *Static Web App* ein.


- Klicken Sie auf **Static Web App (preview)** (Static Web App (Vorschau)) und dann auf die Schaltfläche **Create** (Erstellen):



All services > New >

Static Web App (preview)

Microsoft



Static Web App (preview) Save for later

Microsoft

Create

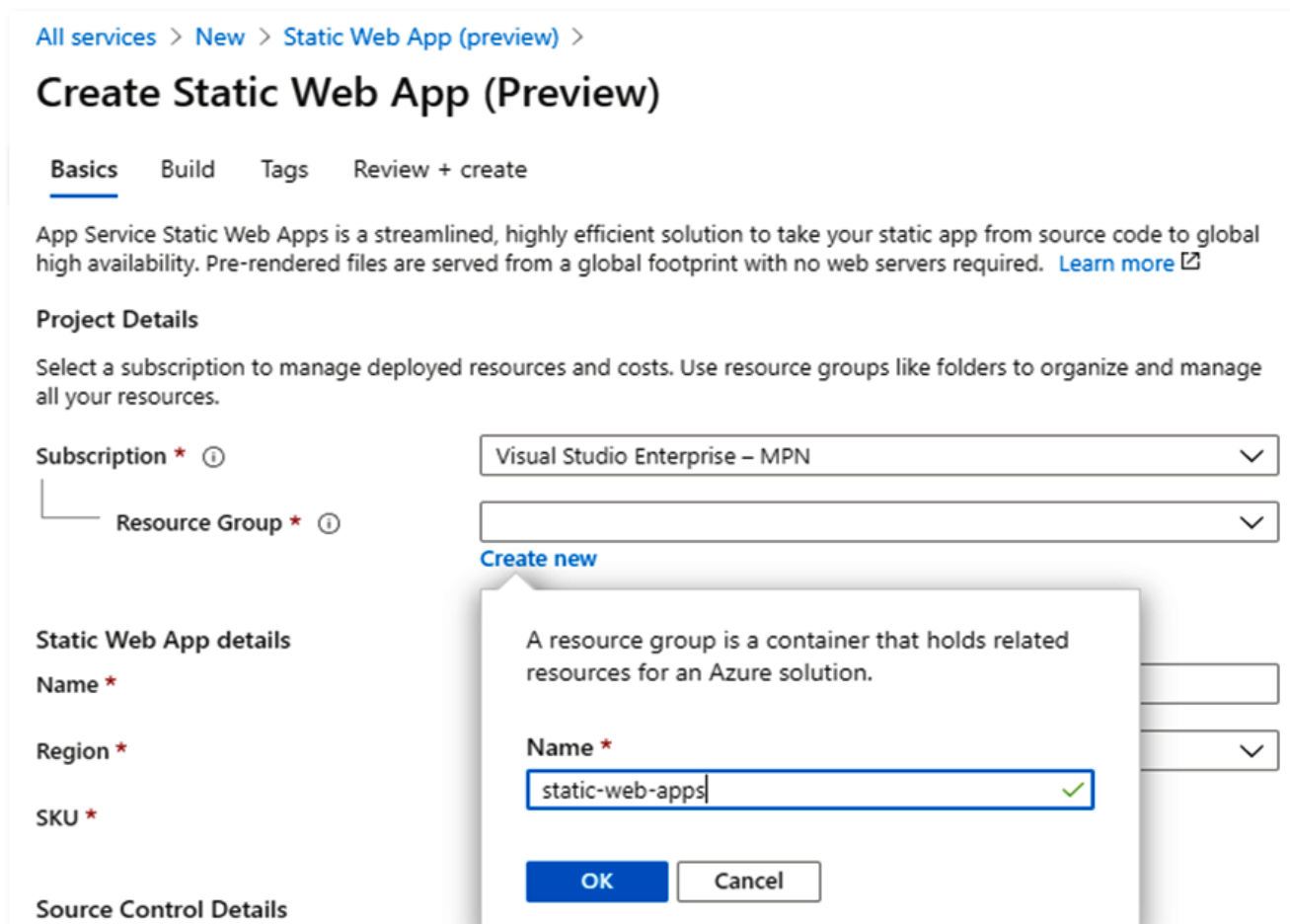
Overview Plans

App Service Static Web Apps is a streamlined, highly efficient solution to take your static web app from source code to global high availability.

Static Web Apps serve pre-rendered files from a global footprint with no web servers required. Static Web Apps development is simple and versatile, designed with React, Angular, Vue, and more in mind. You can even include integrated serverless APIs from Azure Functions.

Abbildung 5: Erstellen einer Static Web App


- Füllen Sie das Formular aus:



All services > New > Static Web App (preview) >



Create Static Web App (Preview)




Basics Build Tags Review + create

App Service Static Web Apps is a streamlined, highly efficient solution to take your static app from source code to global high availability. Pre-rendered files are served from a global footprint with no web servers required. [Learn more](#) 

Project Details


Select a subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.


Subscription *  Visual Studio Enterprise – MPN 

Resource Group *   

[Create new](#)


A resource group is a container that holds related resources for an Azure solution.


Name * 


Name * 

OK **Cancel**

Static Web App details

Name * 

Region * 

SKU * 

Source Control Details


Abbildung 6: Erstellen einer neuen Ressourcengruppe

7. Wählen Sie die Region aus, in der Ihre Azure Static Web App gehostet werden soll:

[All services](#) > [New](#) > [Static Web App \(preview\)](#) >


Create Static Web App (Preview)


Basics Build Tags Review + create

App Service Static Web Apps is a streamlined, highly efficient solution to take your static app from source code to global high availability. Pre-rendered files are served from a global footprint with no web servers required. [Learn more](#) 

Project Details


Select a subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * 

Resource Group * 

[Create new](#)

Static Web App details

Name * 

Region *

SKU *

Source Control Details

GitHub account

- Central US
- East US 2
- East Asia
- West Europe
- West US 2

[Review + create](#)

[< Previous](#)

[Next : Build >](#)

Abbildung 7: Auswahl einer Region

8. Stellen Sie eine Verbindung mit Ihrem GitHub-Konto her:

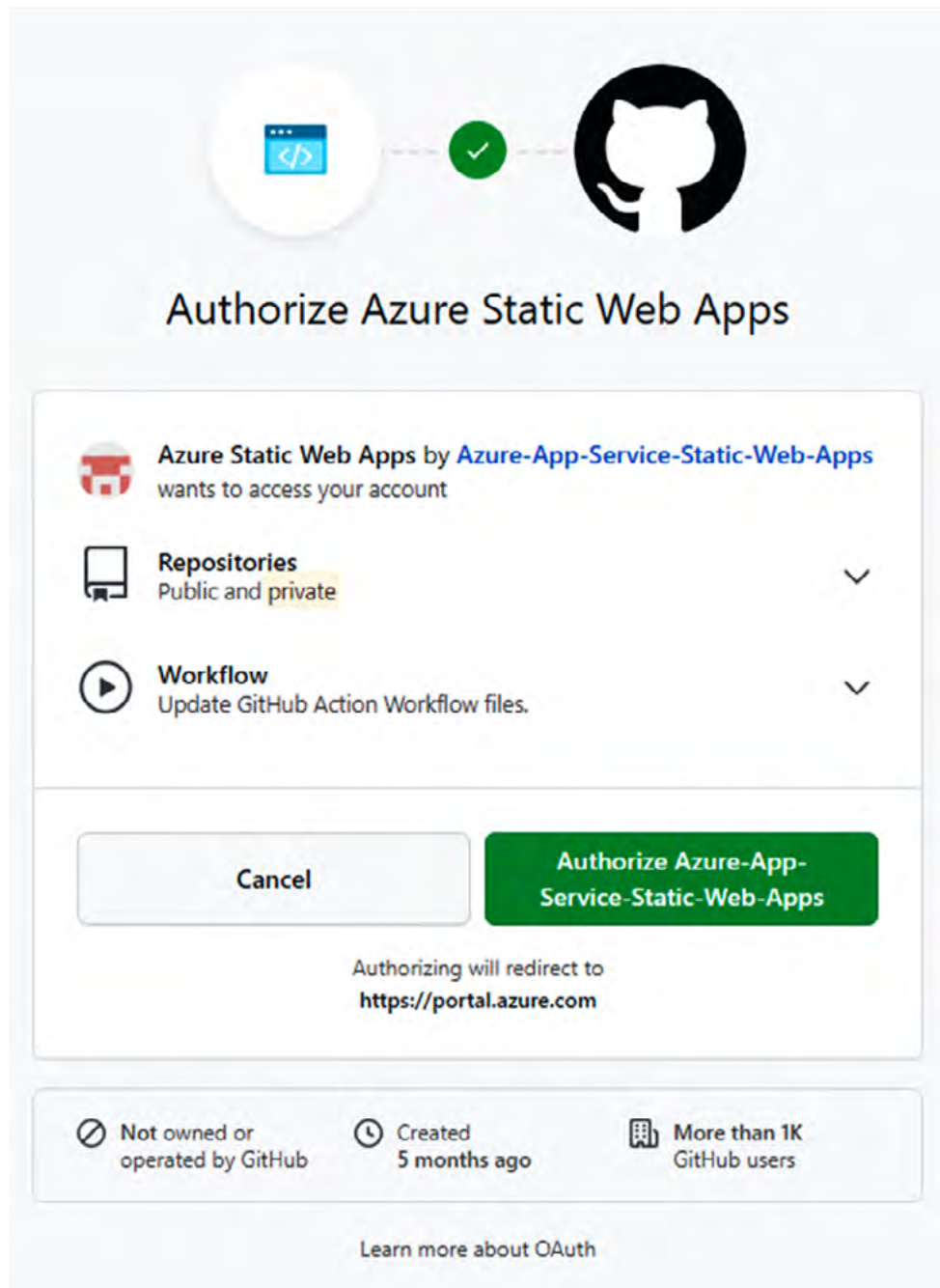


Abbildung 8: Herstellen einer Verbindung mit dem GitHub-Konto

9. Füllen Sie im Abschnitt „GitHub Account“ (GitHub-Konto) die Felder **Organization** (Organisation), **Repository** und **Branch** (Verzweigung) wie folgt aus:

All services > New > Static Web App (preview) >

Create Static Web App (Preview)

Basics Build Tags Review + create

App Service Static Web Apps is a streamlined, highly efficient solution to take your static app from source code to global high availability. Pre-rendered files are served from a global footprint with no web servers required. [Learn more](#)

Project Details

Select a subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * ⓘ Visual Studio Enterprise – MPN

Resource Group * ⓘ (New) static-web-apps
[Create new](#)

Static Web App details

Name * my-first-static-web-app

Region * East US 2

SKU * Free

Source Control Details

GitHub account

If you can't find an organization or repository, you might need to enable additional permissions on GitHub.

Organization *

Repository * my-first-static-web-app

Branch * master

[Review + create](#) < Previous Next : Build >

Abbildung 9: Angaben zu Organisation, Repository und Verzweigung

10. Geben Sie die anfänglichen Build-Variablen wie folgt an:

[All services](#) > [New](#) > [Static Web App \(preview\)](#) >

Create Static Web App (Preview)

Basics **Build** Tags Review + create

Provide initial build variables. These can later be modified in the workflow file.

Build Details

App location * ⓘ

Api location ⓘ

App artifact location ⓘ

[Review + create](#) [< Previous](#) [Next : Tags >](#)

Abbildung 10: Angaben zu den anfänglichen Build-Variablen

11. Wechseln Sie zur Registerkarte **Review + create** (Überprüfen + erstellen), und klicken Sie auf **Create** (Erstellen):

All services > New > Static Web App (preview) >

Create Static Web App (Preview)

Basics Build Tags **Review + create**

Summary

Static Web App
by Microsoft

Details

Subscription	[Redacted]
Resource Group	static-web-apps
Name	my-first-static-web-app
Region	eastus2
SKU	Free
Repository	https://github.com/[Redacted]/my-first-static-web-app
Branch	master
App location	/
API location	api
App artifact location	dist/angular-basic

Create < Previous Next > [Download a template for automation](#)

Abbildung 11: Überprüfen + erstellen

12. Sobald die Bereitstellung abgeschlossen ist, klicken Sie auf **Go to resource** (Zur Ressource), um nach der neuen, für Ihre statische Web-App generierten URL zu suchen:

✓ **Your deployment is complete**

Deployment name: Microsoft.Web-StaticApp-Portal-1dd9adac-8bd2
Subscription: [Visual Studio Enterprise – MPN](#)
Resource group: [static-web-apps](#)

∨ Deployment details [\(Download\)](#)

∧ Next steps

Go to resource

Abbildung 12: Abgeschlossene Bereitstellung

13. Suchen Sie die neue, für Ihre statische Web-App generierte URL:

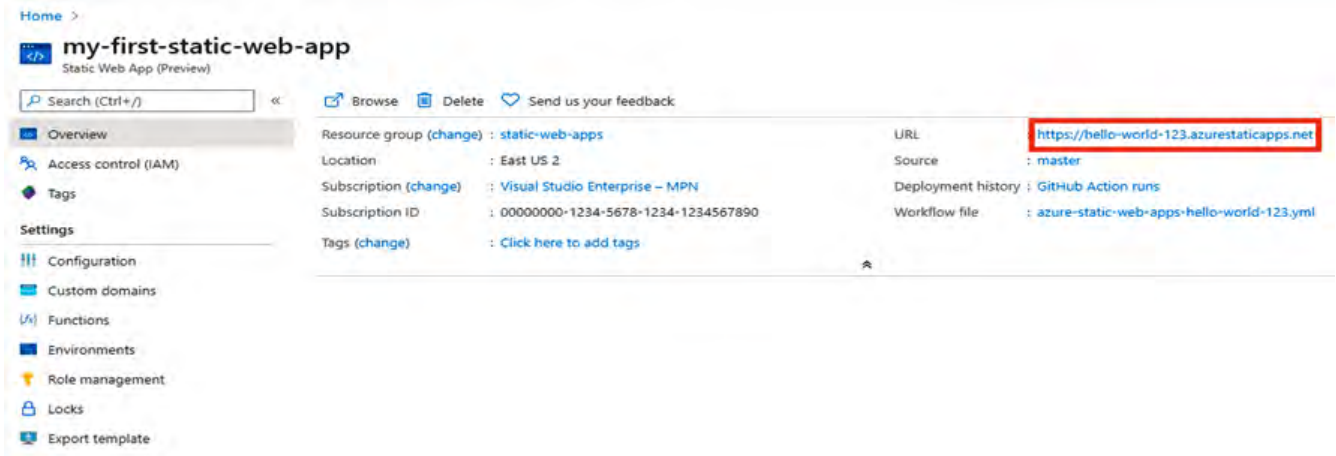


Abbildung 13: URL für Static Web App

14. Öffnen Sie Ihren Browser, und rufen Sie die generierte URL auf, um Ihre statische Web-App in Aktion zu erleben:

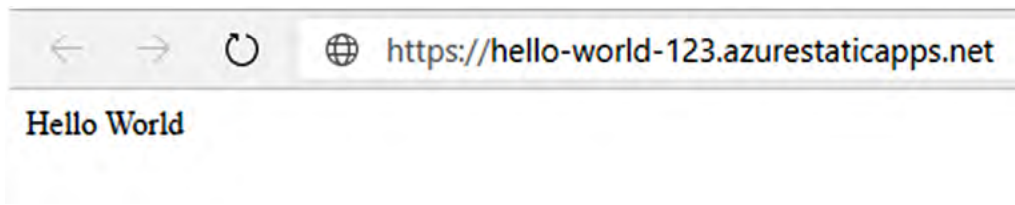


Abbildung 14: Static Web App in Aktion

Herzlichen Glückwunsch, Sie haben Ihre erste Azure Static Web App erfolgreich bereitgestellt.

Weiterführende Informationen

- [Offizielle Dokumentation zu GitHub Actions](#)
- [Überprüfen von Pull Requests in Präproduktionsumgebungen in Azure Static Web Apps](#)

Im nächsten Beispielprojekt werden wir einen intelligenten Chatbot entwickeln.



Kapitel 3: Beispielprojekt – Entwicklung eines intelligenten Chatbots

In diesem Beispielprojekt erfahren Sie, wie sich ein intelligenter Chatbot samt künstlicher Intelligenz (KI) entwickeln lässt. Nach wie vor dominiert KI in Mainstream-Technologien. Deshalb ist es jetzt an der Zeit, dass Entwickler sich deren Leistungsfähigkeit für ihre Anwendungen zunutze machen.

Heutzutage verwenden viele von uns eine Vielzahl von Technologien, um zu kommunizieren. Zum Beispiel:

- Telefonate
- Messaging-Dienste
- Online-Chat-Anwendungen
- E-Mail
- Social-Media-Plattformen
- Tools für die Zusammenarbeit

Wir haben uns an die allgegenwärtige Erreichbarkeit gewöhnt und erwarten, dass Unternehmen problemlos kontaktierbar sind und sofort auf allen gängigen Kanälen reagieren. Darüber hinaus erwarten wir von ihnen, dass sie sich individuell mit uns auseinandersetzen und komplexe Fragen auf persönlicher Ebene beantworten.

Hauptmerkmale von Chatbots

Viele Unternehmen veröffentlichen Supportinformationen sowie Antworten auf häufig gestellte Fragen (FAQs) auf einer Website oder in einer dedizierten App. Dennoch sind Antworten auf spezifische Fragen schwer zu finden. Diese Unternehmen erleben häufig, dass ihre Supportmitarbeiter aufgrund der über die verschiedenen Kanäle (darunter Anrufe, E-Mails, SMS und soziale Medien) eintreffenden Kundenanfragen überlastet sind.

Zahlreiche Firmen entscheiden sich zunehmend für KI-Lösungen, bei denen KI-Agents (gemeinhin als Chatbots bezeichnet) zum Einsatz kommen. Damit lässt sich ein automatisierter First-Level-Support auf allen gängigen Kanäle realisieren. Bots können mit Benutzern Dialoge führen, siehe Abbildung 15: Beispiel für eine Chatbot-Benutzeroberfläche:

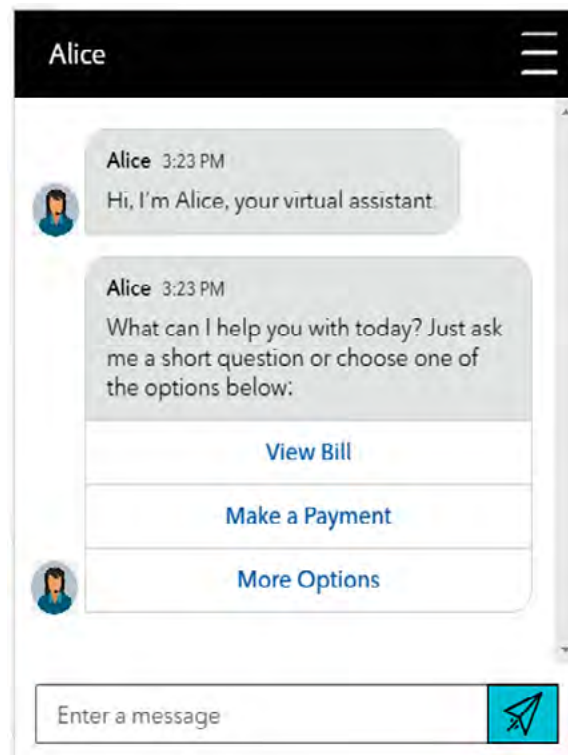


Abbildung 15: Beispiel für eine Chatbot-Benutzeroberfläche

Bei dem hier gezeigten Beispiel handelt es sich um eine Chatbot-Benutzeroberfläche, wie man sie von Einzelhandel-Websites kennt. Bots lassen sich jedoch auch auf mehreren Kanälen einsetzen, einschließlich E-Mail, Social-Media-Plattformen und sogar bei Telefonaten. Unabhängig vom verwendeten Kanal managen Chatbots in der Regel Gesprächsabläufe mithilfe einer Kombination aus natürlicher Sprache und eingeschränkten Auswahlmöglichkeiten, die den Benutzer zu einer Lösung führen.

Anwendungsfälle für Chatbots

In „Abbildung 16: Azure QnA Maker und Azure Bot Service“ sind die beiden wichtigsten Komponenten bei der Erstellung eines intelligenten Chatbot abgebildet. Die erste Komponente ist eine Wissensdatenbank mit Fragen und Antworten. Auf Azure übernimmt QnA Maker diese Funktion. Die zweite Komponente ist ein Bot-Dienst samt Schnittstelle zur Wissensdatenbank. Auf Azure ist dafür der Azure Bot Service zuständig. Mit QnA Maker und Azure Bot Service können Sie einen Chatbot entwickeln, der den Benutzern Antworten auf häufig gestellte Fragen liefert. Als Benutzeroberfläche des Bots dient z. B. eine Chat-Box auf Ihrer Website.

Typischerweise werden die Unterhaltungen in Form von abwechselnd ausgetauschten Nachrichten geführt. Eine der häufigsten Arten des Dialogs ist eine Frage, gefolgt von einer Antwort. Diese Form bildet die Grundlage für viele Support-Bots und basiert häufig auf vorhandenen FAQ-Dokumentationen.

Zwei Schlüsselkomponenten bei der Erstellung eines Proof of Concept für einen intelligenten Chatbot

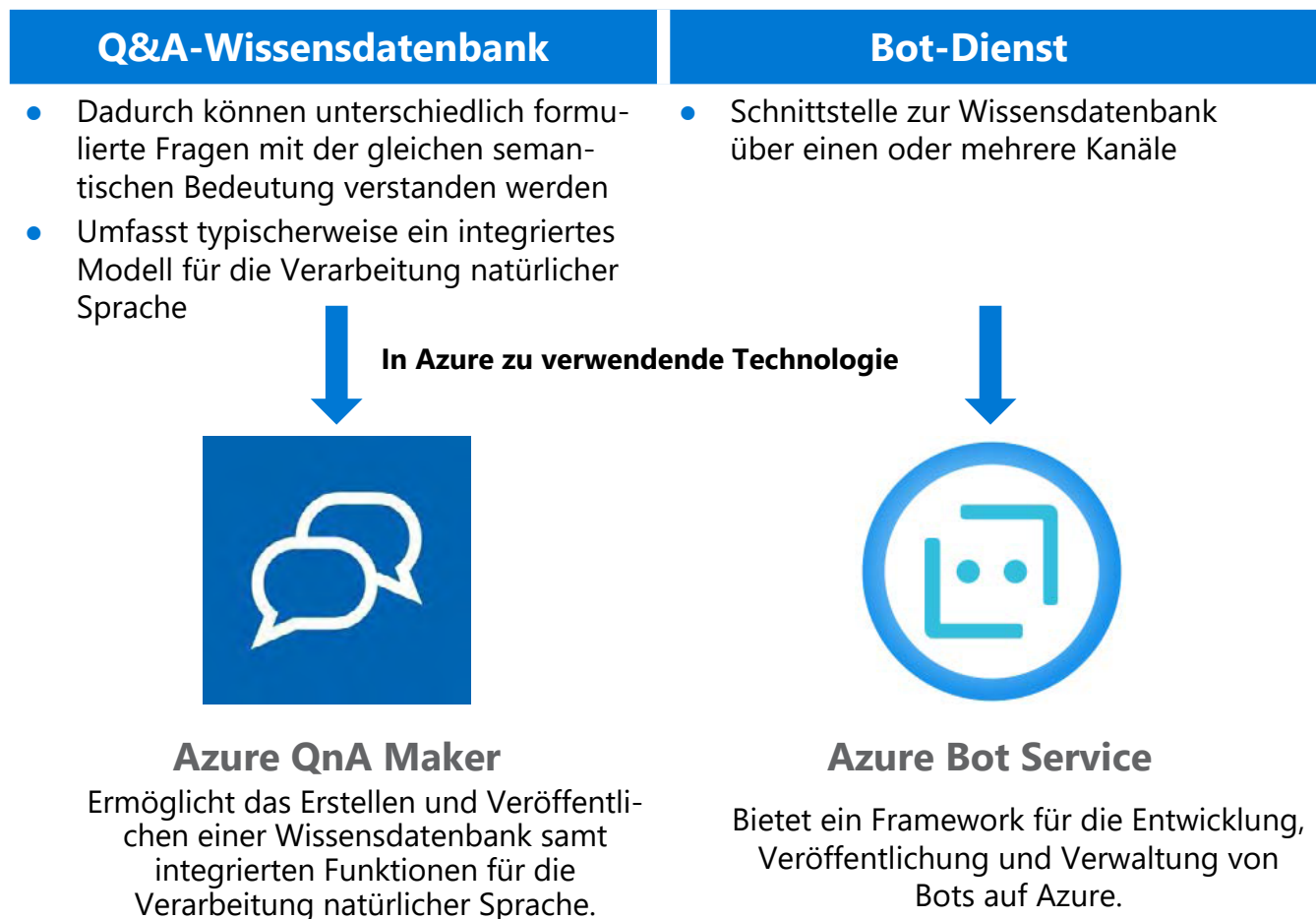


Abbildung 16: Azure QnA Maker und Azure Bot Service

Praktische Umsetzung

1. Melden Sie sich mit Ihren Azure-Anmeldeinformationen beim [QnA Maker-Portal](#) an (wenn Sie noch kein Azure-Abonnement abgeschlossen haben, lesen Sie den Abschnitt „Azure-Abonnement“ in Kapitel 4: Übersicht über Azure für Entwickler).
2. Klicken Sie auf **Create a knowledge base** (Wissensdatenbank erstellen) und dann auf **Create a QnA service** (Q&A-Dienst erstellen), um eine neue QnA Maker-Ressource zu erstellen:

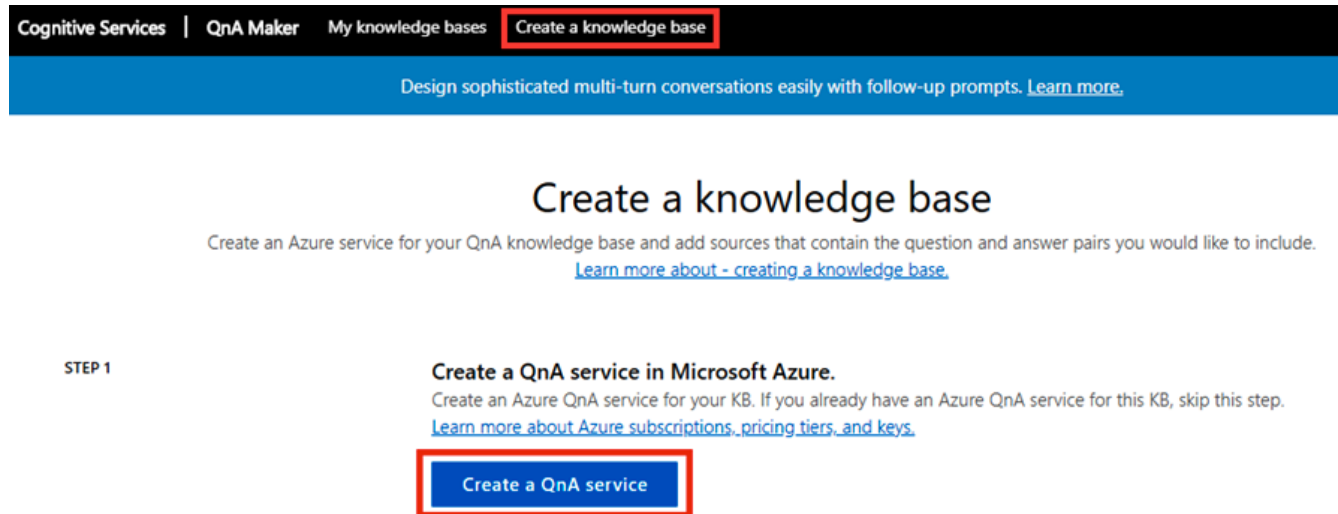


Abbildung 17: Erstellen einer neuen QnA Maker-Ressource

- Nachdem das Azure-Portal gestartet wurde, füllen Sie das Formular wie folgt aus, und klicken Sie dann auf die Registerkarte **Review + create** (Überprüfen + erstellen). Die Eingabe im Feld **Name** muss global eindeutig sein. Wenn Sie eine Fehlermeldung („Namenskonflikt“) erhalten, versuchen Sie es einfach noch einmal mit einem anderen eindeutigen Namen:

Home >

Create

QnA Maker

***Basics** Tags Review + create

QnA Maker is a cloud-based API service that lets you create a conversational question-and-answer layer over your existing data. Use it to build a knowledge base by extracting questions and answers from your semi-structured content, including FAQs, manuals, and documents. Answer users' questions with the best answers from the QnAs in your knowledge base—automatically. Your knowledge base gets smarter, too, as it continually learns from user behavior. [Learn more](#)

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * ⓘ Microsoft Azure MVP

Resource group * ⓘ (New) my-qnamaker-poc
[Create new](#)

Name * ⓘ my-qnamaker-poc

Pricing tier * ⓘ Free F0 (3 managed documents per month, 3 transactions per second...)

Azure Search details - for data

When you create a QnAMaker resource, you host the data in your own Azure subscription. Azure Search is used to index your data.

Azure Search location * (US) East US

Azure Search pricing tier * ⓘ Free F (3 Indexes)

App Service details - for runtime

When you create a QnAMaker resource, you host the runtime in your own Azure subscription. App Service is the compute engine that runs the QnA Maker queries for you.

App name * ⓘ my-qnamaker-poc

Website location * (US) East US

Review + create Next : Tags >

Abbildung 18: Angaben auf der Registerkarte Basics (Grundlagen)

4. Nachdem Ihre QnA Maker-Ressource bereitgestellt wurde, wechseln Sie zurück zur Seite [Create a knowledge base](#) (Wissensdatenbank erstellen), und aktualisieren Sie sie. Fahren Sie anschließend mit Schritt 2 fort:

STEP 2

Connect your QnA service to your KB.

After you create an Azure QnA service, refresh this page and then select your Azure service using the options below

Refresh

* Microsoft Azure Directory ID

* Azure subscription name

* Azure QnA service

* Language

Abbildung 19: Schritt 2 von QnA Maker – Q&A-Dienst mit Wissensdatenbank verbinden

5. In Schritt 3 geben Sie Ihrer Wissensdatenbank einen Namen:

STEP 3

Name your KB.

The knowledge base name is for your reference and you can change it at anytime.

* Name

Abbildung 20: Schritt 3 von QnA Maker – Wissensdatenbank benennen

6. In Schritt 4 befüllen Sie die Datenbank wie folgt:

STEP 4

Populate your KB.
 Extract question-and-answer pairs from an online FAQ, product manuals, or other files. Supported formats are .tsv, .pdf, .doc, .docx, .xlsx, containing questions and answers in sequence. [Learn more about knowledge base sources.](#) Skip this step to add questions and answers manually after creation. The number of sources and file size you can add depends on the QnA service SKU you choose. [Learn more about QnA Maker SKUs.](#)

Enable multi-turn extraction from URLs, .pdf or .docx files. [Learn more.](#)

* Default answer text ?

Sorry, I don't have an answer for you.

URL

<https://docs.microsoft.com/azure/cognitive-services/qnamaker/troubleshooting>

+ Add URL

File name

+ Add file

Chit-chat

Give your bot the ability to answer thousands of small-talk questions in a voice that fits your brand. When you add chit-chat to your knowledge base by selecting a personality below, the questions and responses will be automatically added to your knowledge base, and you'll be able to edit them anytime you want. [Learn more about chit-chat.](#)

None
 Professional
 Friendly
 Witty
 Caring
 Enthusiastic

Abbildung 21: Schritt 4 von QnA Maker – Wissensdatenbank befüllen

7. In Schritt 5 klicken Sie auf **Create your KB** (Wissensdatenbank erstellen):

STEP 5

Create your KB
 The tool will look through your documents and create a knowledge base for your service. If you are not using an existing document, the tool will create an empty knowledge base table which you can edit.

Create your KB

Abbildung 22: Schritt 5 von QnA Maker – Wissensdatenbank erstellen

8. Wählen Sie im QnA Maker-Portal auf der Seite **Edit** (Bearbeiten) die Option **+ Add QnA pair** (+ Q&A-Paar hinzufügen) auf der Symbolleiste aus:

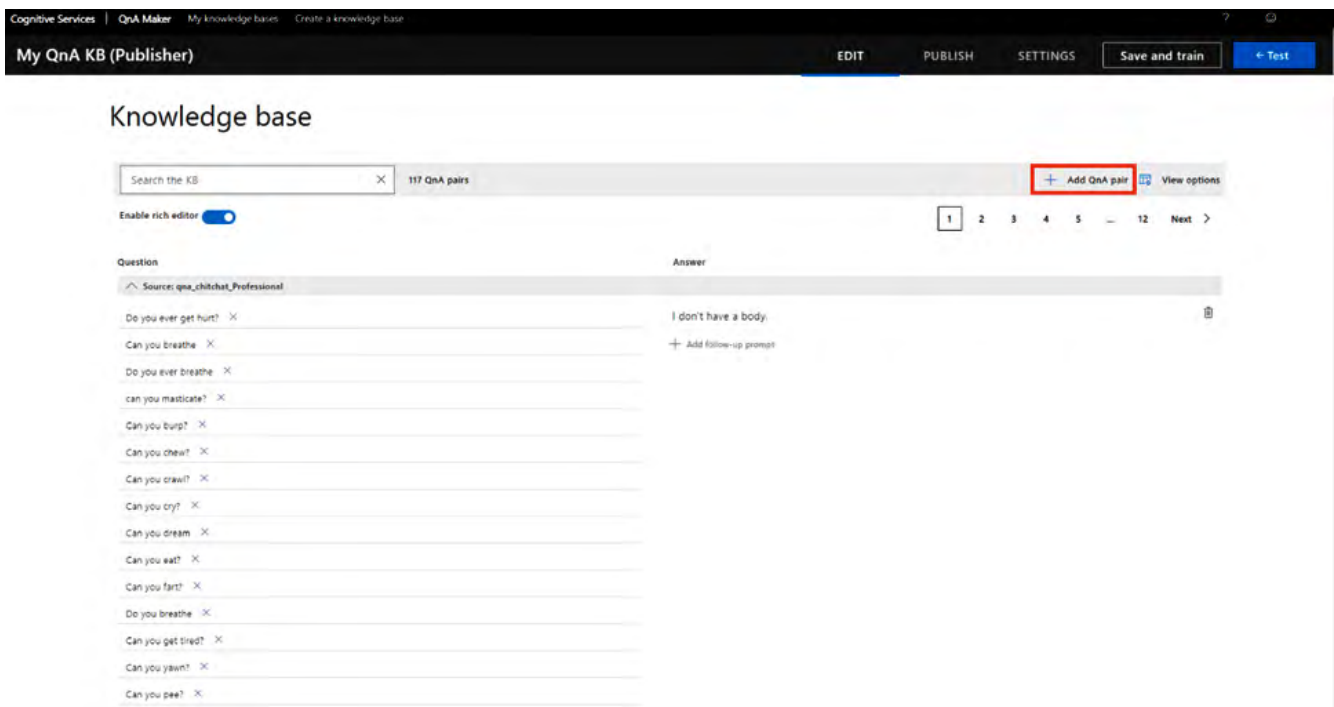


Abbildung 23: Hinzufügen eines Q&A-Paars

9. Fügen Sie die Frage und die Antwort hinzu. Klicken Sie dann auf die Schaltfläche **Save and train** (Speichern und trainieren):

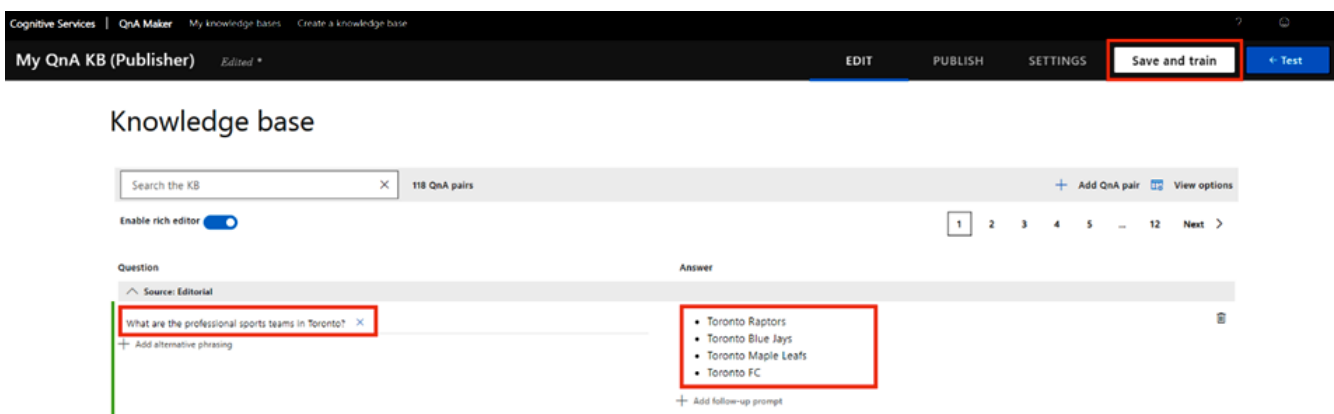


Abbildung 24: Hinzufügen von Fragen und Antworten

10. Sie können Ihre Wissensdatenbank sofort testen, indem Sie auf **Test** (Testen) klicken. Geben Sie anschließend eine Frage ein, z. B. „*What are the professional sports teams in Toronto*“ (Welche Sportmannschaften gibt es in Toronto):

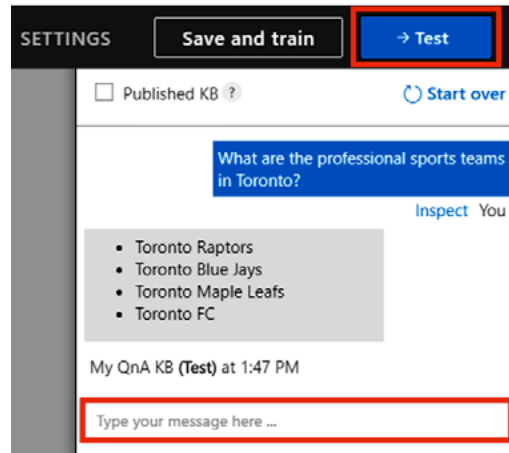


Abbildung 25: Testen der Wissensdatenbank

11. Klicken Sie auf **Inspect** (Inspizieren), um die Details der Unterhaltung anzuzeigen:

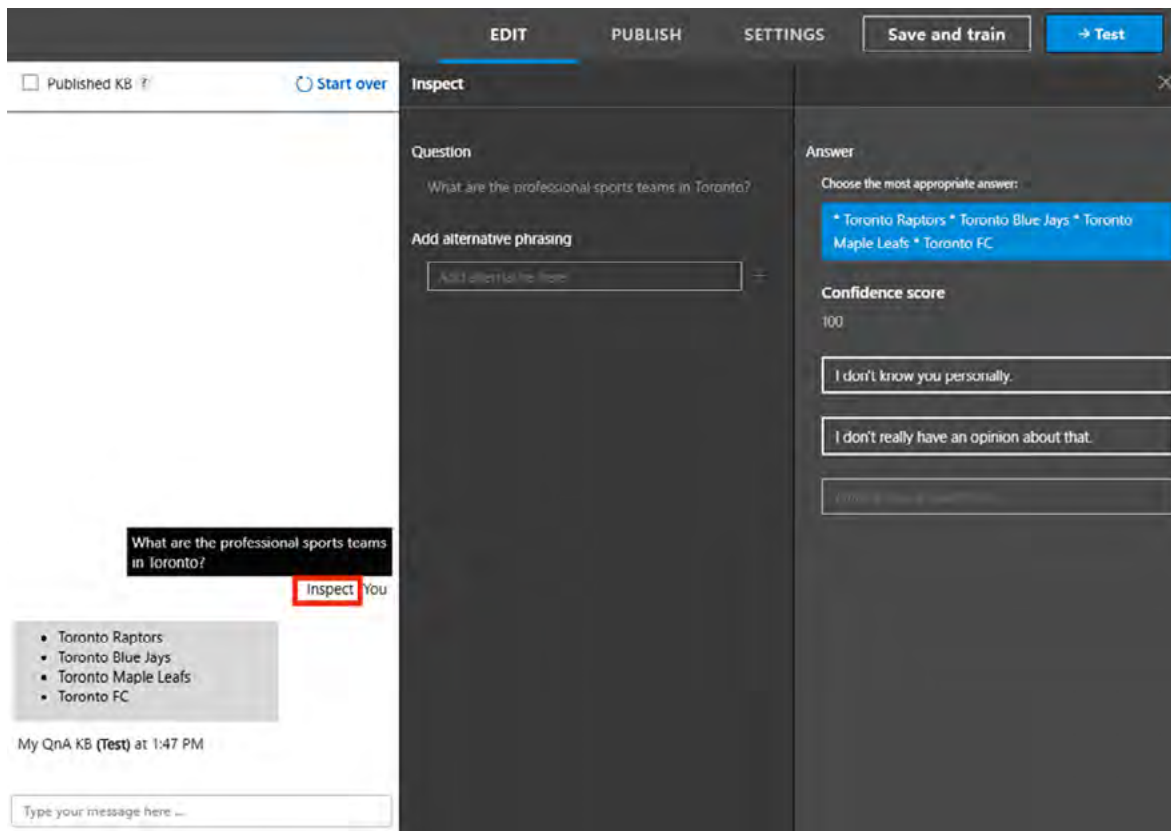


Abbildung 26: Untersuchen der Einzelheiten der Unterhaltung

12. Klicken Sie auf die Schaltfläche **Test** (Testen), um den Testbereich zu schließen.

Q&A-Wissensdatenbank veröffentlichen

Wenn Sie die Wissensdatenbank veröffentlichen, werden deren Inhalte vom Test- zu einem Produktionsindex nach Azure Search verschoben.

Klicken Sie im QnA Maker-Portal auf **Publish** (Veröffentlichen):

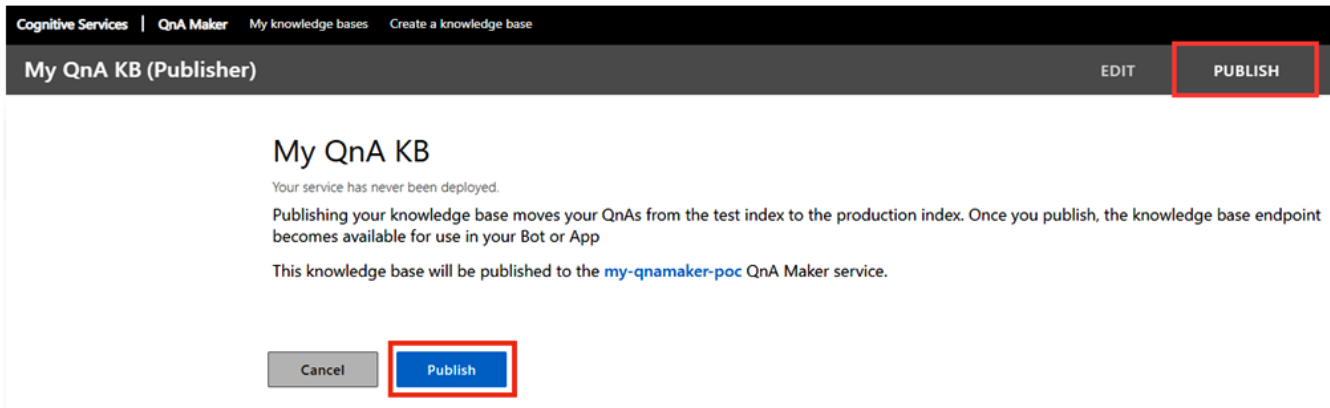
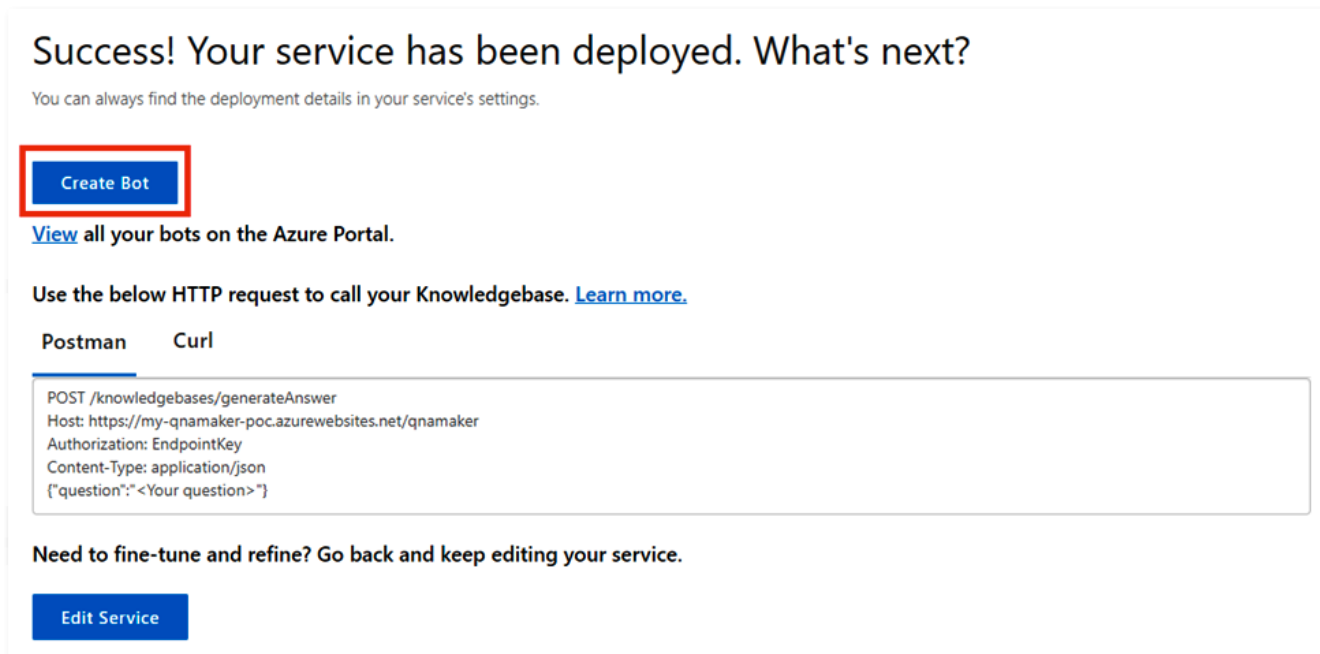


Abbildung 27: Veröffentlichen der Q&A-Wissensdatenbank

Bot in Azure Bot Service erstellen

Als Nächstes erstellen Sie einen Bot in Azure Bot Service, der mit der zuvor erstellten Wissensdatenbank verbunden wird:

1. Nachdem die Wissensdatenbank erfolgreich bereitgestellt wurde, klicken Sie auf **Create Bot** (Bot erstellen), um die Azure Bot Service-Seite im Azure-Portal aufzurufen:



Success! Your service has been deployed. What's next?

You can always find the deployment details in your service's settings.

[Create Bot](#)

[View](#) all your bots on the Azure Portal.

Use the below HTTP request to call your Knowledgebase. [Learn more.](#)

Postman Curl

```
POST /knowledgebases/generateAnswer
Host: https://my-qnamaker-poc.azurewebsites.net/qnamaker
Authorization: EndpointKey
Content-Type: application/json
{"question": "<Your question>"}
```

Need to fine-tune and refine? Go back and keep editing your service.

[Edit Service](#)

Abbildung 28: Aufrufen der Azure Bot Service-Seite

2. Füllen Sie auf der Azure Bot Service-Seite das Formular wie folgt aus:

Home >

Web App Bot

Bot Service

Bot handle * ⓘ

Subscription *

 ▾

Resource group *

 ▾

[Create new](#)

Location * ⓘ

 ▾

Pricing tier (View full pricing details)

 ▾

App name * ⓘ

 ✓
.azurewebsites.net

SDK language *

C# Node.js

QnA Auth Key * ⓘ

***App service plan/Location** >

my-qnamaker-poc/East US

Application Insights ⓘ

On Off

Microsoft App ID and password ⓘ >

Auto create App ID and password

Create Automation options

Abbildung 29: Hinzufügen von Web App Bot-Details

3. Sobald der Bot bereitgestellt wurde, öffnen Sie ihn über Bot Services:

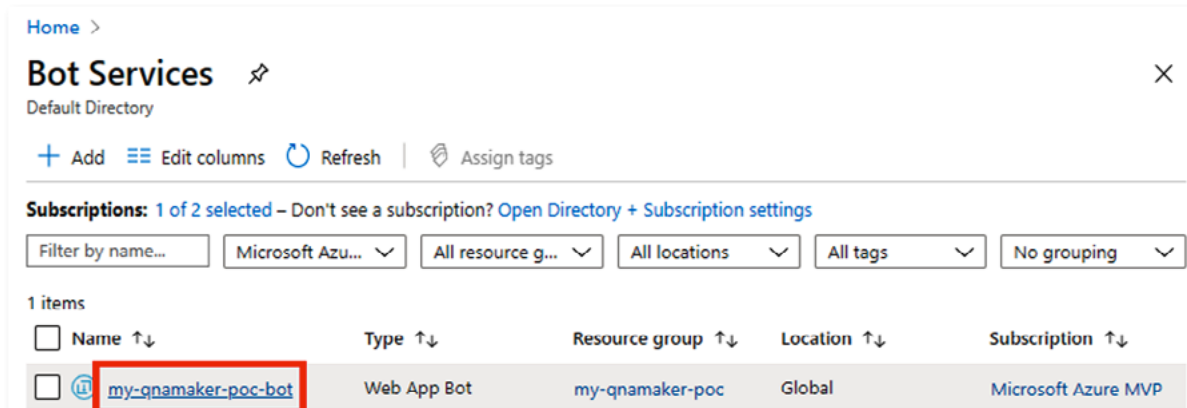


Abbildung 30: Öffnen des Bots über Bot Services

4. Wählen Sie unter **Bot management** (Bot-Verwaltung) die Option **Test in Web Chat** (In Web Chat testen) aus:

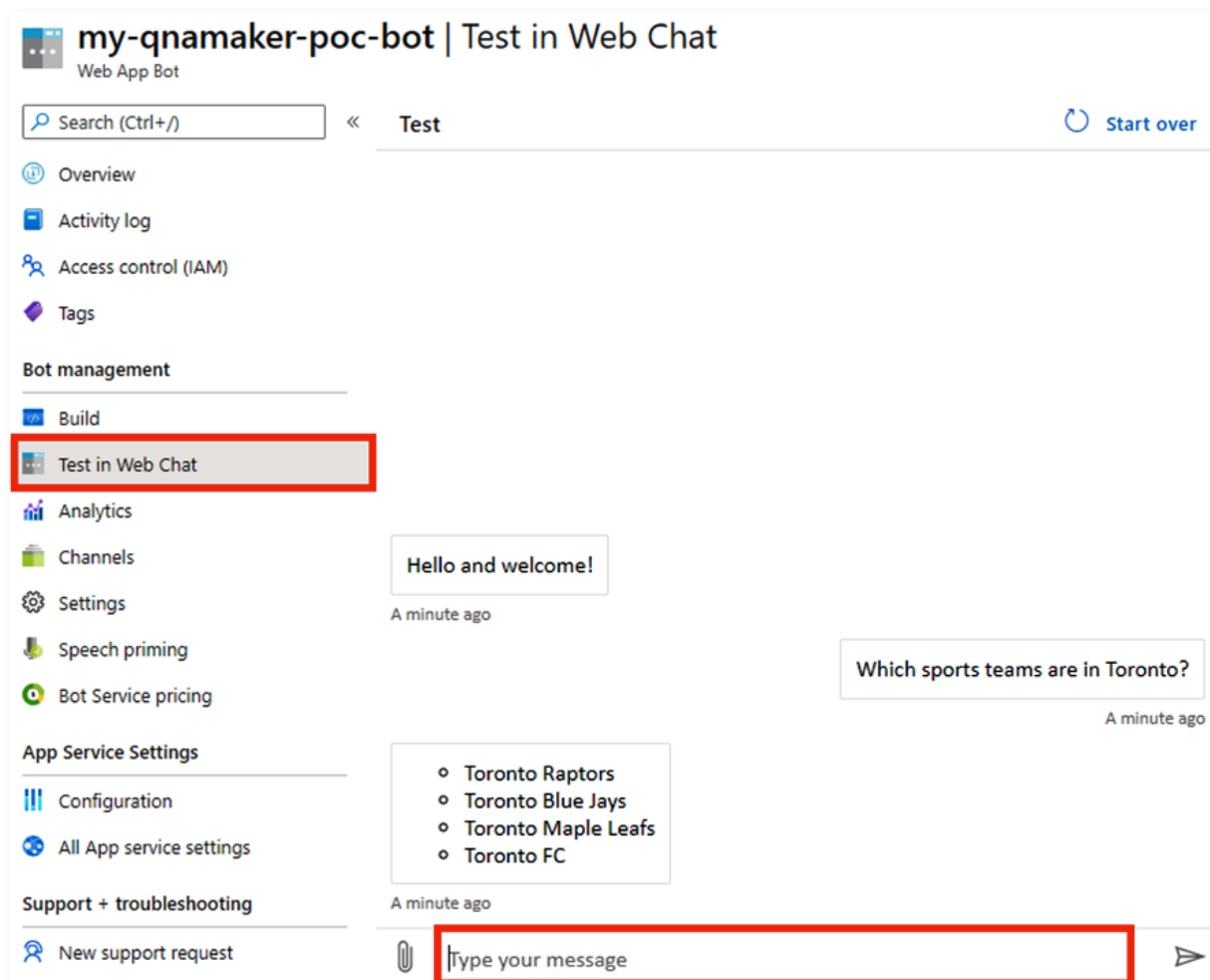


Abbildung 31: Auswahl der Option „Test in Web Chat“

Zusammenfassend lässt sich sagen, dass Sie mithilfe des QnA Maker eine neue Wissensdatenbank in Azure erstellt haben. Sie haben eine öffentliche URL zur Wissensdatenbank hinzugefügt. Danach haben Sie Ihr eigenes Q&A-Paar hinzugefügt, trainiert und getestet. Nach dem Veröffentlichen Ihrer Wissensdatenbank haben Sie einen Web App Bot in Azure Bot Services erstellt. Abschließend haben Sie den Bot in Azure getestet. Bots stellen automatisierten Support auf mehreren Kommunikationskanäle bereit und helfen so, Supportkosten zu senken. Wir haben Ihnen gezeigt, wie Sie mithilfe von QnA Maker und Azure Bot Services eine Chatbot erstellen, der Fragen von Benutzern beantworten kann.

Damit endet unsere Tour durch zwei praktische Projekte. Wir hoffen, Sie damit inspiriert zu haben, Ihre eigenen Proof-of-Concept-Projekte zu erstellen. Im nächsten Abschnitt geben wir eine Übersicht über Azure für Entwickler. Außerdem erfahren Sie, wie sich das geeignete Cloudmodell für Ihre Azure-Proof-of-Concept-Projekte auswählen lässt.



Kapitel 4: Übersicht über Azure für Entwickler

Erste Schritte mit Microsoft Azure

Wie in der Einführung erläutert, ist Azure eine Cloudplattform. Damit ist Folgendes möglich:

- Hosten Ihrer vorhandenen Anwendungen mit minimalen Änderungen
- Entwicklung neuer Cloudanwendungen

Es gibt eine Vielzahl von einsatzbereiten Azure-Diensten, die Sie in Ihre Anwendungen integrieren können. So profitieren Sie umgehend von den neuen Cloudfunktionen, ohne derartige Dienste selbst entwickeln zu müssen. In diesem Abschnitt wird Folgendes erläutert:

- Die Vorteile von Azure
- Erste Schritte mit Azure
- Tools, die für die Entwicklung von Anwendung für Azure nötig sind
- So wählen Sie das geeignete Cloudmodell für Ihr Azure-Proof-of-Concept-Projekt

Die Vorteile von Azure

Im Folgenden sind einige wichtige Vorteile aufgeführt, die das Hosten von Anwendungen auf Azure bietet:

Geringere Kosten

- Durch die Entwicklung von Anwendungen auf Azure spart Ihr Unternehmen dank des nutzungsbasierten Zahlungsmodells langfristig Investitionskosten.
- Sie bezahlen nur für die Dienste, die Sie tatsächlich nutzen.
- Kapitalausgaben für den Kauf von On-Premises-Servern entfallen.

Geringer Wartungsaufwand

- Dank verwalteter Dienste können Sie sich auf die Entwicklung Ihrer Anwendung konzentrieren.
- Azure übernimmt für Sie die Wartung der Infrastruktur.

Skalierbarkeit

- Azure-Dienste sind hochgradig skalierbar, um den wachsenden Anforderungen Ihres Unternehmens gerecht zu werden.

Hochverfügbarkeit und Zuverlässigkeit

- Sie können Ihre Azure-Dienste so konfigurieren, dass Ihre Anwendungen hoch verfügbar und zuverlässig sind.

Azure-Abonnement

Sie benötigen ein Azure-Abonnement, um mit der Entwicklung von Anwendungen für Azure beginnen zu können. Falls Ihr Unternehmen Ihnen Azure-Guthaben bereitstellt, sind Sie startklar. Andernfalls können Sie sich auch für ein [kostenfreies Azure-Konto](#) registrieren.

Was umfasst das kostenfreie Azure-Konto?

Mit dem kostenfreien Azure-Konto erhalten Sie für 12 Monate kostenfreien Zugriff auf Dienste sowie eine Gutschrift in Höhe von 200 USD, um Azure 30 Tage lang testen zu können. Die Angebote können sich im Laufe der Zeit ändern. Aktuelles zum kostenfreien Azure-Konto finden Sie auf der zugehörigen [FAQ-Seite](#).

Für Azure-Neueinsteiger bieten wir zahlreiche kostenfreie Ressourcen für einen reibungslosen Einstieg. Diese Ressourcen finden Sie in Kapitel 5: Weiterführende Lernmaterialien und Ressourcen.

Der nächste Abschnitt befasst sich mit den Tools, die Sie für das Entwerfen Ihres Proof-of-Concept-Projekts für Azure benötigen.

Die für das Entwerfen eines Azure-Proof-of-Concept-Projekts nötigen Tools

In der folgenden Liste sind Tools aufgeführt, die für die in diesem Leitfaden gezeigten Beispiele unerlässlich sind:

- [Azure-Abonnement](#)
- [Visual Studio Code](#)
- [Azure Functions Core Tools](#)

- [GitHub-Konto](#)
- [Microsoft Edge-Browser \(basierend auf Chromium\)](#)
- [Node.js](#)

Im nächsten Abschnitt erfahren Sie, wie Sie das geeignete Cloudmodell für Ihr Azure-Proof-of-Concept-Projekt auswählen.

So wählen Sie das geeignete Cloudmodell für Ihr Azure-Proof-of-Concept-Projekt

In „Kapitel 1: Proof-of-Concept-Leitfaden“ haben Sie etwas über den Proof of Concept erfahren. Sobald Sie Ihren Proof of Concept geplant haben, können Sie mit der Implementierung und den Tests beginnen. In diesem Abschnitt zeigen wir Ihnen die vier für Ihr Azure-Proof-of-Concept-Projekt zur Auswahl stehenden Cloudmodelle (siehe Abbildung 32: Azure-Cloudmodelle für das Entwickeln und Hosten von Anwendungen). Wir beschreiben Anwendungsfälle für jedes der Cloudmodelle, damit Sie eine fundierte Entscheidung treffen können.

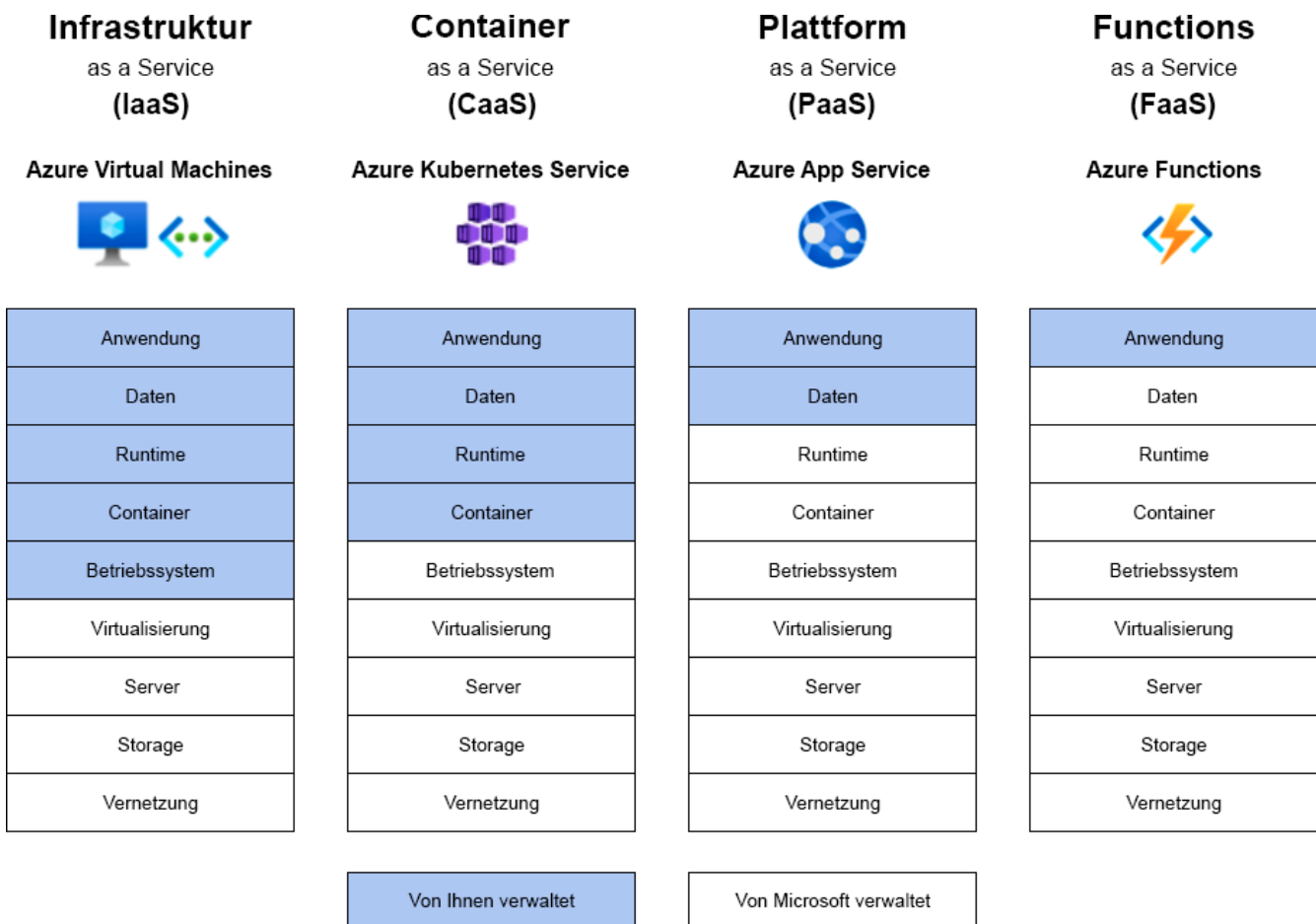


Abbildung 32: Azure-Cloudmodelle für das Entwickeln und Hosten von Anwendungen

Infrastructure-as-a-Service (IaaS)

Beim IaaS-Modell gewährt Ihnen Azure die volle Kontrolle über die Verwaltung Ihrer eigenen Anwendungshosting-Umgebung.

Einführung in Azure Virtual Machines

Durch das IaaS-Modell ermöglicht Azure die Bereitstellung oder Migration Ihrer Anwendung zu virtuellen Windows- oder Linux-Maschinen. So haben Sie die volle Kontrolle über die Computerkonfiguration. Beim IaaS-Modell sind Sie jedoch für alle Betriebssystemupgrades sowie für die Installation, Konfiguration und Wartung der Serversoftware verantwortlich.

Hauptmerkmale von Azure Virtual Machines

- Mit Azure Virtual Machines haben Sie die volle Kontrolle über das Betriebssystem.
- Sie können das Verhältnis zwischen CPU und Arbeitsspeicher, die Computerfamilie (SKU) und die Datenträgerlayouts genau abstimmen.

Anwendungsfälle für Azure Virtual Machines

- Azure Virtual Machines ist ideal, wenn Sie die volle Kontrolle über Ihre Anwendungsinfrastruktur haben möchten oder Workloads von On-Premises-Anwendungen zu Azure migrieren wollen, ohne große Änderungen vornehmen zu müssen.

Weiterführende Informationen

- [Windows Virtual Machines-Dokumentation](#)
- [Linux Virtual Machines-Dokumentation](#)

Container-as-a-Service (CaaS)

Ein Container ist per Definition eine Standardeinheit im Bereich der Software. Er bündelt eine gesamte Laufzeitumgebung (eine Anwendung, die für deren Ausführung erforderlichen Konfigurationsdateien und alle ihre Abhängigkeiten, Bibliotheken und sonstigen Binärdateien) in einem einzigen Paket. Durch die Containerisierung der Anwendungsplattform und ihrer Abhängigkeiten werden Abweichungen in den Betriebssystemumgebungen und der zugrunde liegenden Infrastruktur davon abstrahiert.

Einführung in Azure Kubernetes Service

Wenn Sie im CaaS-Modell mehr als einen Container ausführen, benötigen Sie einen Orchestrator. Bei Azure heißt der verwaltete Container-Orchestrator Azure Kubernetes Services (AKS).

AKS verwaltet Ihre gehostete Kubernetes-Umgebung und vereinfacht die Bereitstellung eines verwalteten Kubernetes-Clusters auf Azure. Sie können einen AKS-Cluster im Azure-Portal (mittels Azure-Befehlszeilenschnittstelle) oder vorlagengesteuerte Bereitstellungsoptionen (z. B. Azure Resource Manager-Vorlagen und Terraform) erstellen. Mit AKS lassen sich containerisierte Anwendungen problemlos bereitstellen und verwalten. Sie müssen kein Experte in der Container-Orchestrierung sein, um AKS verwenden zu können.

Bei der Bereitstellung eines AKS-Clusters werden der Kubernetes-Master und alle Worker-Knoten für Sie bereitgestellt und konfiguriert. Während der Bereitstellung können auch andere Funktionen (z. B. Azure Active Directory-Integration, Überwachung und erweiterte Netzwerkdienste) konfiguriert werden. AKS unterstützt zudem Windows Server-Container.

Hauptmerkmale von AKS

- Einfachere, weniger aufwendige Verwaltung von Kubernetes, da ein Großteil der Aufgaben an Azure ausgelagert wird.
- Azure übernimmt wichtige Aufgaben wie die Systemüberwachung und -wartung für Sie.
- Azure verwaltet für Sie die Kubernetes-Master.
- Kubernetes-Master sind gebührenfrei.
- Sie verwalten und warten nur die Agent-Knoten.
- Sie zahlen nur für die Agent-Knoten innerhalb Ihres Clusters.

Anwendungsfälle für AKS

- AKS ist eine gute Wahl, wenn Sie die Bereitstellung und Verwaltung von auf Microservices basierenden Anwendungen vereinfachen möchten. Mit AKS lassen sich außerdem vorhandene Anwendungen zu Containern migrieren und diese auszuführen.

Weiterführende Informationen

- [Azure Kubernetes Service](#)

Platform-as-a-Service (PaaS)

Im Rahmen des PaaS-Modell bietet Azure eine vollständig verwaltete gehostete Umgebung, auf der Ihre Anwendung ausgeführt werden kann. Sie müssen sich keine Gedanken über die zugrunde liegenden Infrastruktureinheiten machen.

Einführung in Azure App Service

Azure App Service ist ein PaaS, mit dem Sie Ihre webbasierten Projekte schnellstmöglich veröffentlichen können. Mit App Service können Sie Ihre Web-Apps ganz einfach erweitern, um Ihre mobilen Clients zu unterstützen und REST-APIs zu veröffentlichen.

Mit App Service können Sie die folgenden Arten von Anwendungen erstellen:

- Web-Apps
- APIs
- Back-Ends von mobilen Apps

Da alle diese Anwendungstypen die App Service-Laufzeit gemeinsam nutzen, können Sie aus einem einzelnen Projekt oder einer einzelnen Lösung eine Website hosten, mobile Clients unterstützen und Ihre APIs auf Azure bereitstellen.

App Service wurde mit Blick auf DevOps entwickelt. Es unterstützt verschiedene Tools für die Veröffentlichung und für CI/CD, z. B.:

- Azure DevOps
- GitHub
- Bitbucket
- Docker Hub
- Azure Container Registry

Hauptmerkmale von Azure App Service

- Authentifizierung mithilfe von Social-Media-Anbietern (Microsoft-Konto, Facebook, Twitter, Google usw.)
- Automatische Skalierung basierend auf Datenverkehr
- Testen in der Produktionsumgebung
- Kontinuierliche und containerbasierte Bereitstellungen

Anwendungsfälle für Azure App Service

- Azure App Service unterstützt eine breite Palette von Webtechnologien
- Ideal, um einen Proof of Concept für fast jeden Webdienst oder Web-Front-End zu erstellen (darunter .NET, Java, Python, PHP usw.)
- Azure App Service ist ideal, wenn:
 - Sie vorhandene Webanwendungen zu Azure migrieren möchten.
 - Sie eine vollständig verwaltete Hosting-Plattform für Ihre Webanwendungen benötigen.
 - Sie eine containerisierte Webanwendung bereitstellen und ausführen müssen.
 - Sie REST-APIs in Ihrer App verfügbar machen müssen.

Zu Azure App Service migrieren

- Das Migrate to Azure App Service-Tool unterstützt Sie bei der Migration von bestehenden .NET- und PHP-Anwendungen zu Azure App Service.

Weiterführende Informationen

- [Azure App Service: Übersicht](#)

Praktische Umsetzung

Über die Seite [Azure App Service testen](#) erhalten Sie Zugriff auf App Service.

Auf diese Weise können Sie kurzfristig eine Proof-of-Concept-App bereitstellen und die Plattform in einer Sandbox-Umgebung testen, ohne dass dafür ein Azure-Abonnement nötig wäre. Es ist kostenfrei und unverbindlich.

1. Auf der Hauptseite haben Sie die Wahl zwischen **Web App** (Web-App) oder **Web App for Containers** (Web-App für Container). Wählen Sie **Web App** (Web-App) aus:

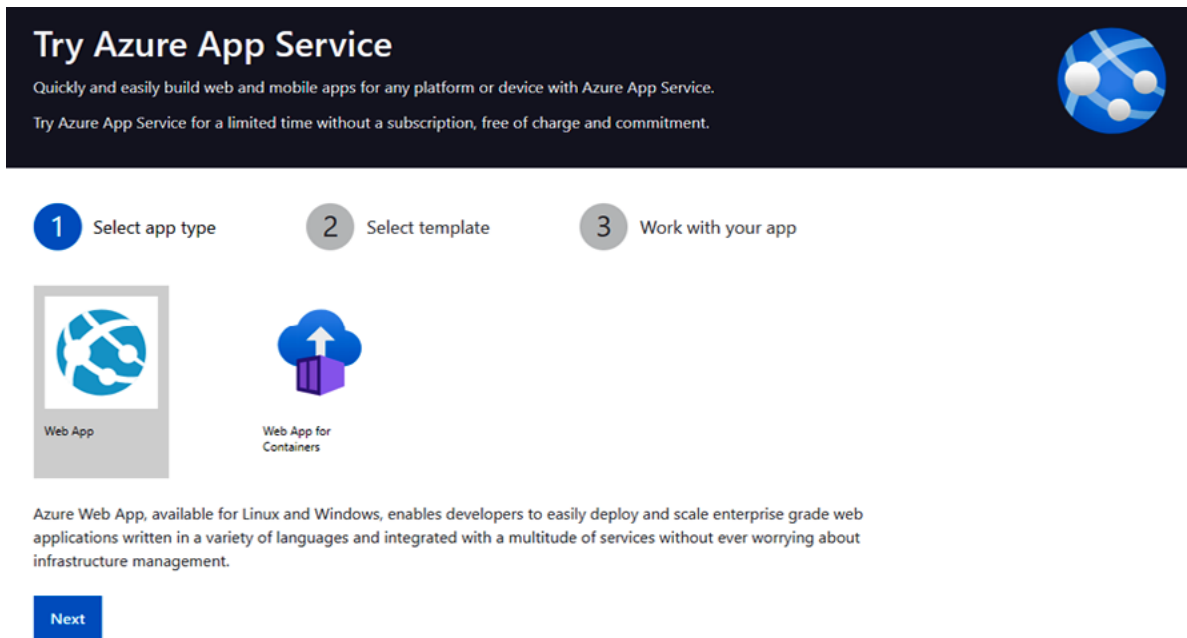


Abbildung 33: Zur Auswahl stehende Optionen „Web App“ und „Web App for Containers“

2. Entscheiden Sie sich als Nächstes für eine Sprache, und wählen Sie eine Vorlage aus, um Ihre Web-App zu erstellen. Wählen Sie „C#“ und dann „ASP.NET Core“ aus. Klicken Sie dann auf **Create** (Erstellen):

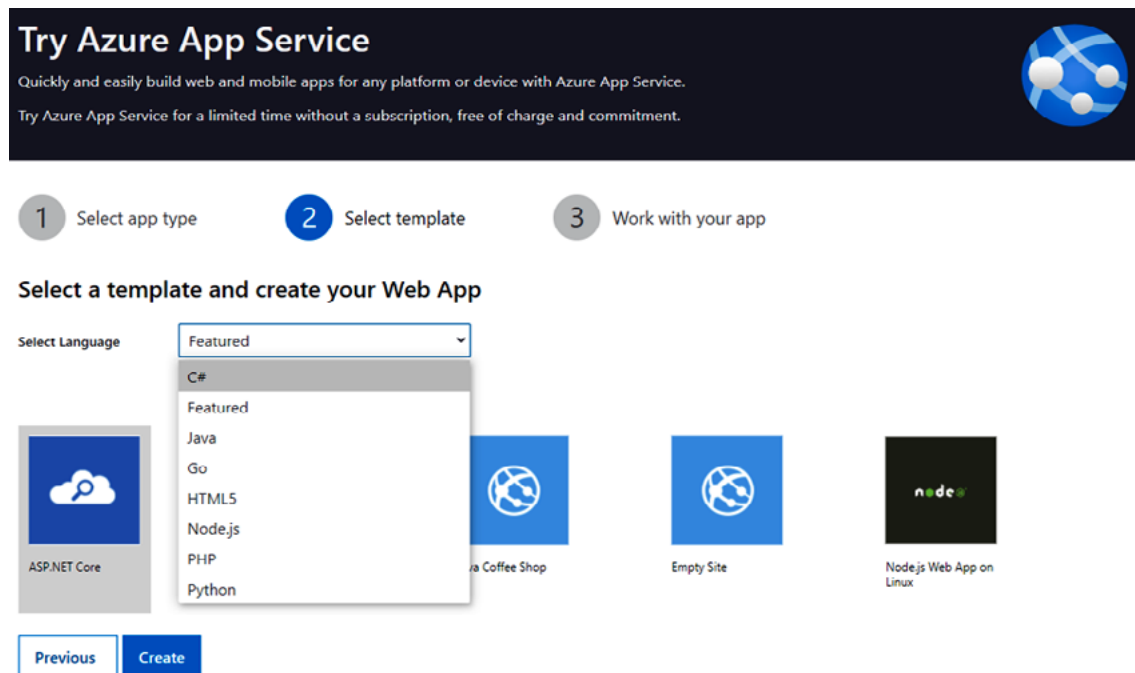


Abbildung 34: Auswahl einer Sprache

3. Im nächsten Schritt erfolgt die Anmeldung. Sie haben die Wahl zwischen vier Optionen:

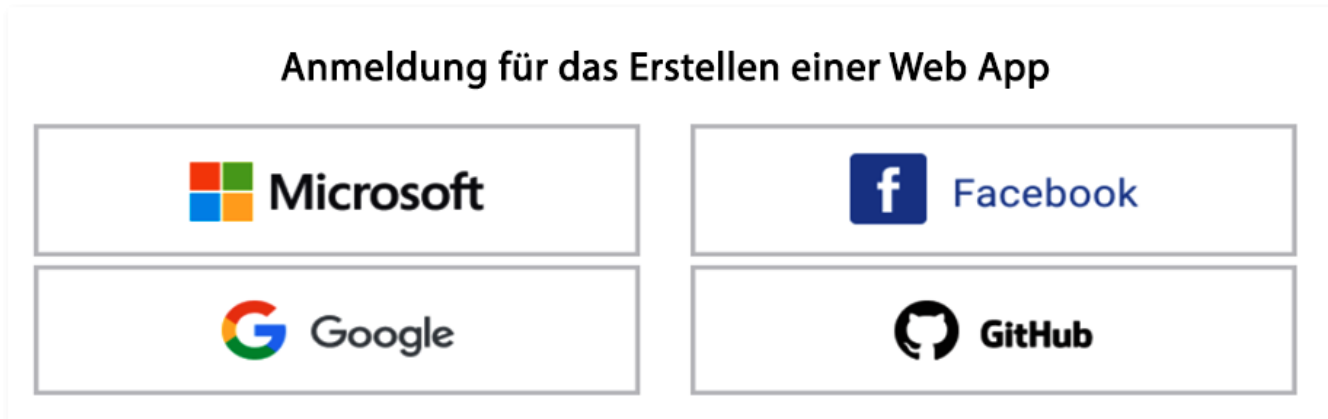


Abbildung 35: Anmeldung für das Erstellen einer Web App

4. Nach erfolgreicher Bereitstellung können Sie Änderungen an den Inhalten vornehmen und mit den geänderten Resultaten experimentieren:

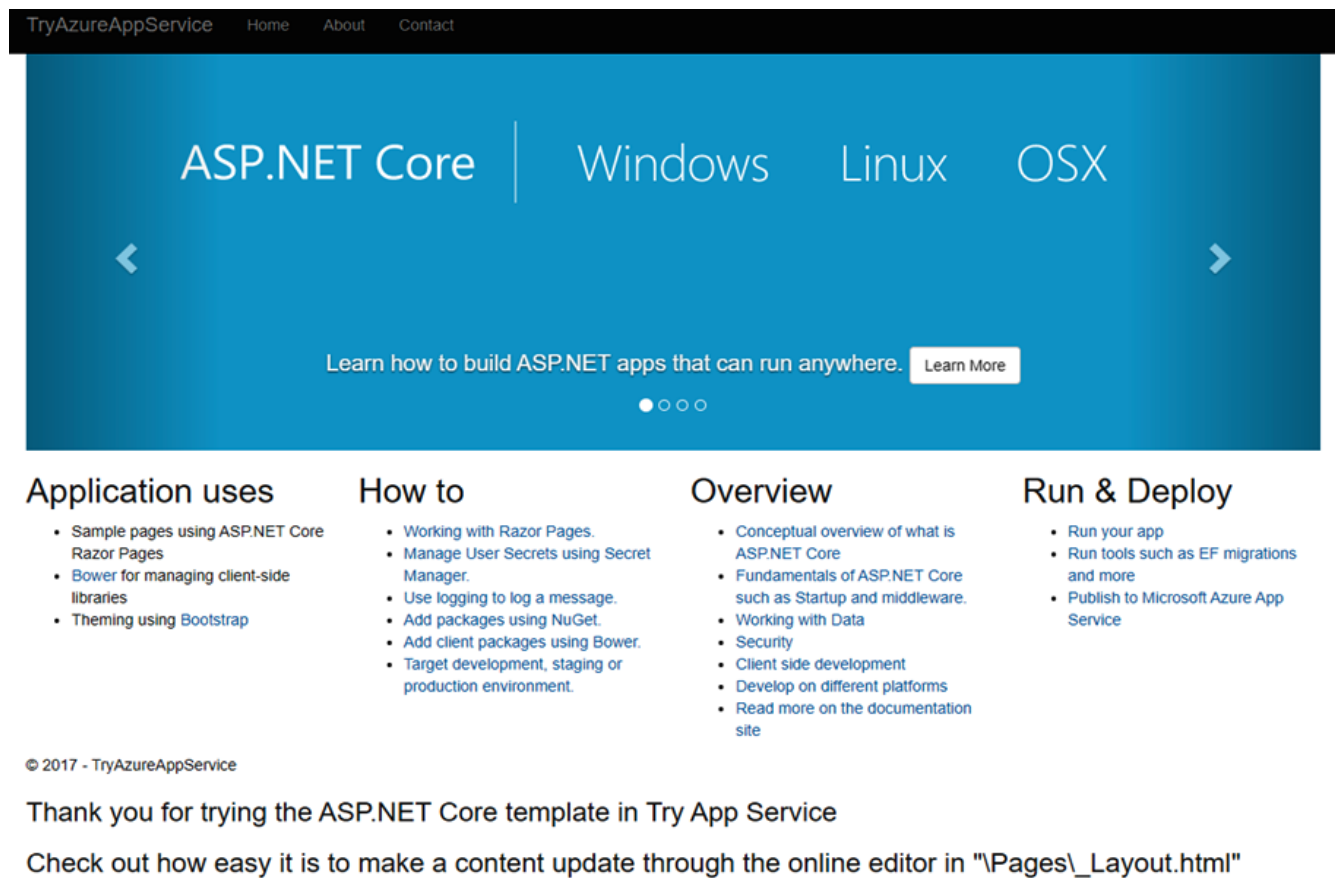


Abbildung 36: Erfolgreich bereitgestellte Website

5. Wenn Sie mit der Proof-of-Concept-App fertig sind, können Sie zurück zur Website [Azure App Service testen](#) wechseln und mit einer anderen Vorlage experimentieren.

Als Nächstes betrachten wir das serverlose Angebot, bekannt als Azure Functions.

Function-as-a-Service (FaaS)

Das Faas-Modell von Azure bietet Ihnen eine serverlose Umgebung. Sie müssen nur Ihren Code entwickeln. Um die zugrunde liegenden Details bei der Infrastruktur kümmert sich Azure.

Einführung in Azure Functions

Mit Azure Functions können Sie serverlosen Code ausführen, ohne Ihre eigene Infrastruktur bereitstellen zu müssen. Eine Azure-Funktion ist eine Einheit der Codelogik, die durch eine HTTP-Anfrage, ein Ereignis in einem anderen Azure-Dienst oder basierend auf einem Zeitplan ausgelöst werden kann.

Azure Functions ist serverlos, damit Sie sich auf das Schreiben von Code konzentrieren können, ohne sich um einen Server für das Ausführen des Codes kümmern zu müssen. Die Abrechnung erfolgt erst, wenn der Endpunkt aufgerufen wird. Wenn die Endpunkte nicht verwendet werden, fallen keine Gebühren an. Bei der nutzungsbasierten Abrechnung zahlen Sie nur für die Zeit, in der Ihr Code ausgeführt wird, wobei Azure nach Bedarf skaliert. Dadurch ist Azure Functions die ideale Wahl für APIs.

Hauptmerkmale von Azure Functions

Merkmale	Vorteile
Keine Notwendigkeit, eine Infrastruktur zu verwalten	Dadurch können Sie sich darauf konzentrieren, einen Mehrwert zu schaffen. Die Skalierung kann automatisiert und flexibel erfolgen.
Unterstützt viele beliebte Sprachen	Unterstützte Sprachen: <ul style="list-style-type: none"> - C# - JavaScript - F# - Java - PowerShell - Python - TypeScript <p>Weitere Informationen zu den in Azure Functions unterstützten Sprachen finden Sie in dieser Dokumentation.</p>

Merkmale	Vorteile
Vollständige Entwicklungsumgebung	Integrierte Tools und DevOps-Funktionen für das Entwickeln, Debuggen, Bereitstellen und Überwachen.
Vereinfachte Integration	Einfache Integration in Azure-Dienste und SaaS-Angebote.
Nutzungsbasierte Preise	Beim nutzungsbasierten Hosting-Modell wird Ihnen nur dann etwas berechnet, wenn Ihr Code ausgeführt wird.

Dank Azure Functions können Sie kleine Funktionseinheiten schnell entwickeln und in einer flexiblen Umgebung hosten, die die Skalierung automatisch verwaltet.

Anwendungsfälle für Azure Functions

- Azure Functions ist eine gute Wahl, wenn Sie über Code verfügen, der von anderen Azure-Diensten, von webbasierten Ereignissen oder gemäß einem Zeitplan ausgelöst wird. Azure Functions ist auch dann nützlich, wenn Sie den Aufwand eines vollständig gehosteten Projekts scheuen oder wenn Sie nur für die Zeit bezahlen wollen, in der Ihr Code ausgeführt wird.
- Zur besseren Veranschaulichung nehmen wir Folgendes an: Sie möchten, dass die Größe von Bilddateien nach dem Hochladen in den Azure Blob Storage automatisch geändert wird. Dazu erstellen Sie eine Azure Function, die jedes Mal aktiviert wird, wenn eine neue Bilddatei in den Azure Blob Storage hochgeladen wird. Die Funktion passt anschließend die Bildgröße an und überträgt die Datei zurück zum Blob Storage-Konto. Die Infrastruktur für die Verbindung mit dem Blob Storage muss nicht programmiert werden, Sie müssen sie lediglich konfigurieren.

Weiterführende Informationen

- [Azure Functions-Dokumentation](#)
- [Konzepte für Azure Functions-Trigger und -Bindungen](#)

Praktische Umsetzung

Ebenso wie den Azure App Service können Sie Azure Functions ohne ein Azure-Abonnement kostenfrei in einer Sandbox-Umgebung testen:

1. Rufen Sie [diese URL](#) auf, und erstellen Sie Ihre erste Azure-Funktion.
2. Aktivieren Sie das Kontrollkästchen **I'm not a robot** (Ich bin kein Roboter), und klicken Sie auf **Create a sample function app** (Beispiel-Funktions-App erstellen):

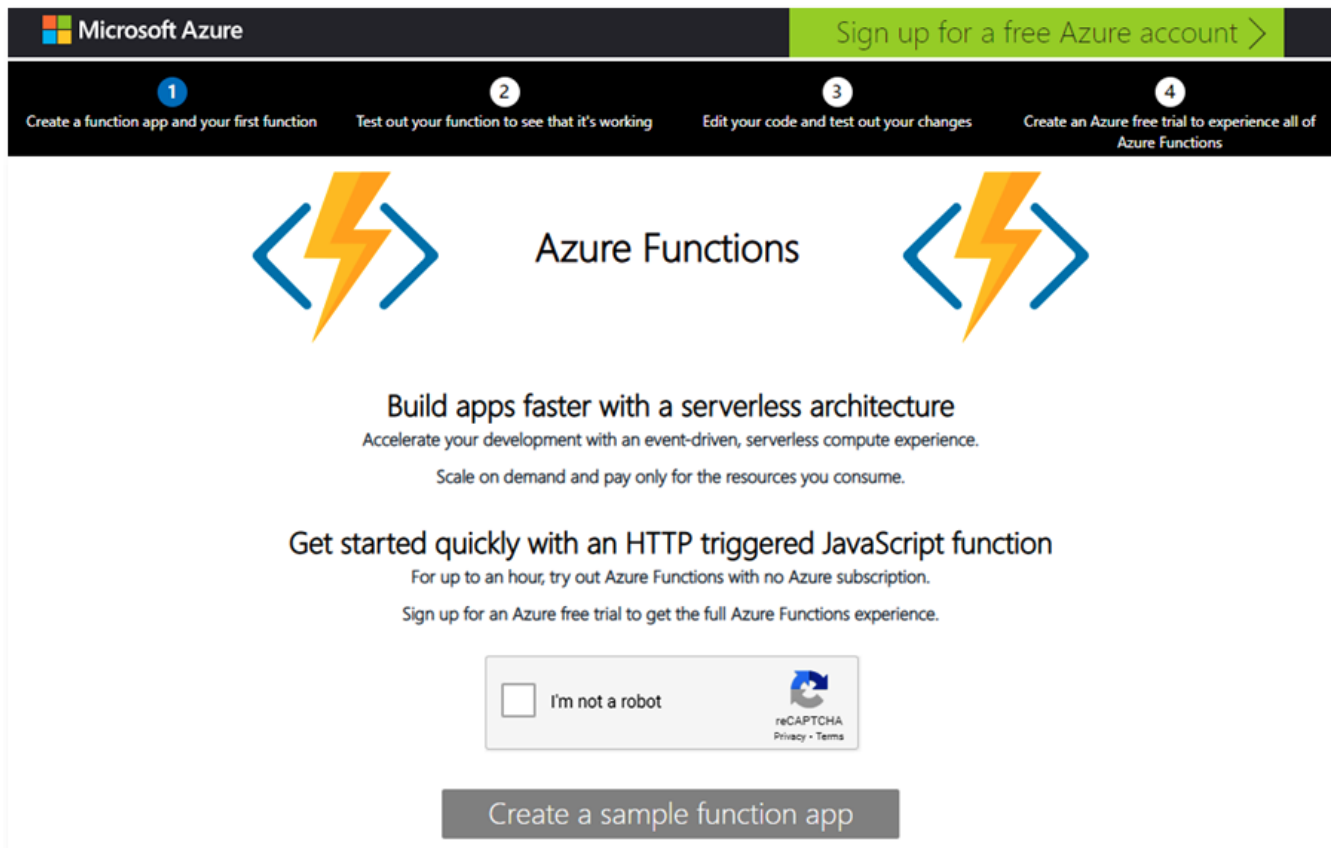


Abbildung 37: Erstellen einer Funktions-App als Beispiel

3. Im Editorbereich liegt bereits eine HTTP-Trigger-Funktions-App als Beispiel vor. Die Funktion wird durch eine HTTP-GET- oder HTTP-POST-Anfrage ausgelöst und sendet als Ausgabe eine HTTP-Antwort auf Grundlage des bereitgestellten Benutzercodes. Klicken Sie auf **Run** (Ausführen), um sie in Aktion zu erleben:

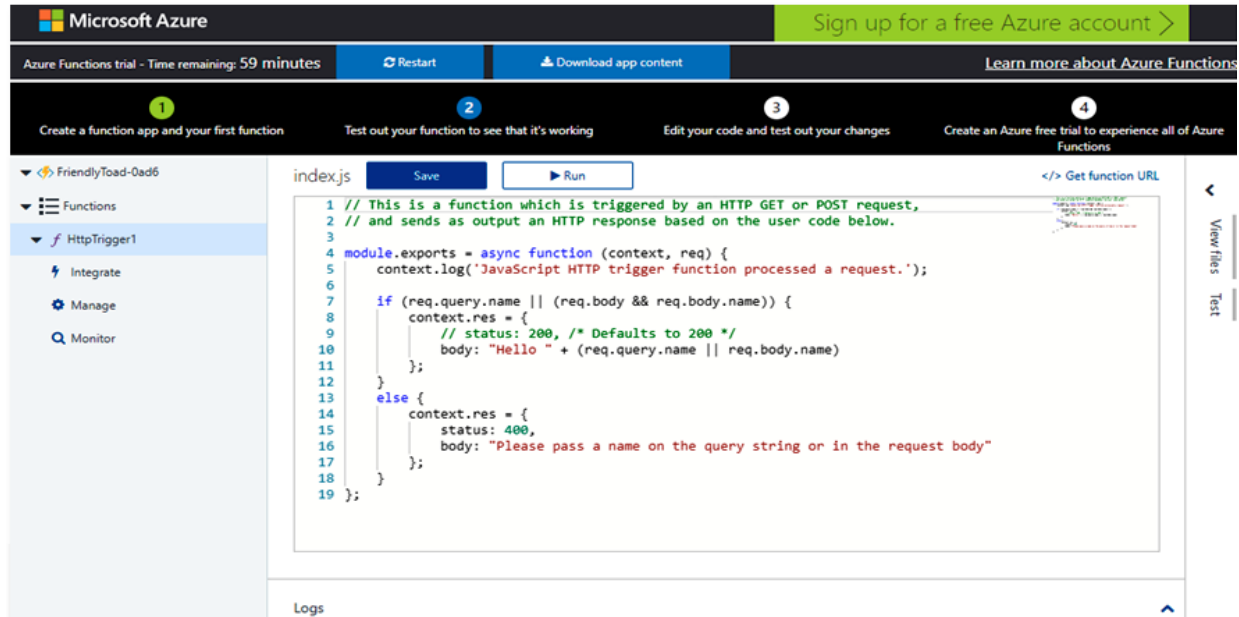


Abbildung 38: Schaltfläche „Run“ (Ausführen)

4. Nach erfolgreicher Ausführung wird die folgende Ausgabe angezeigt:

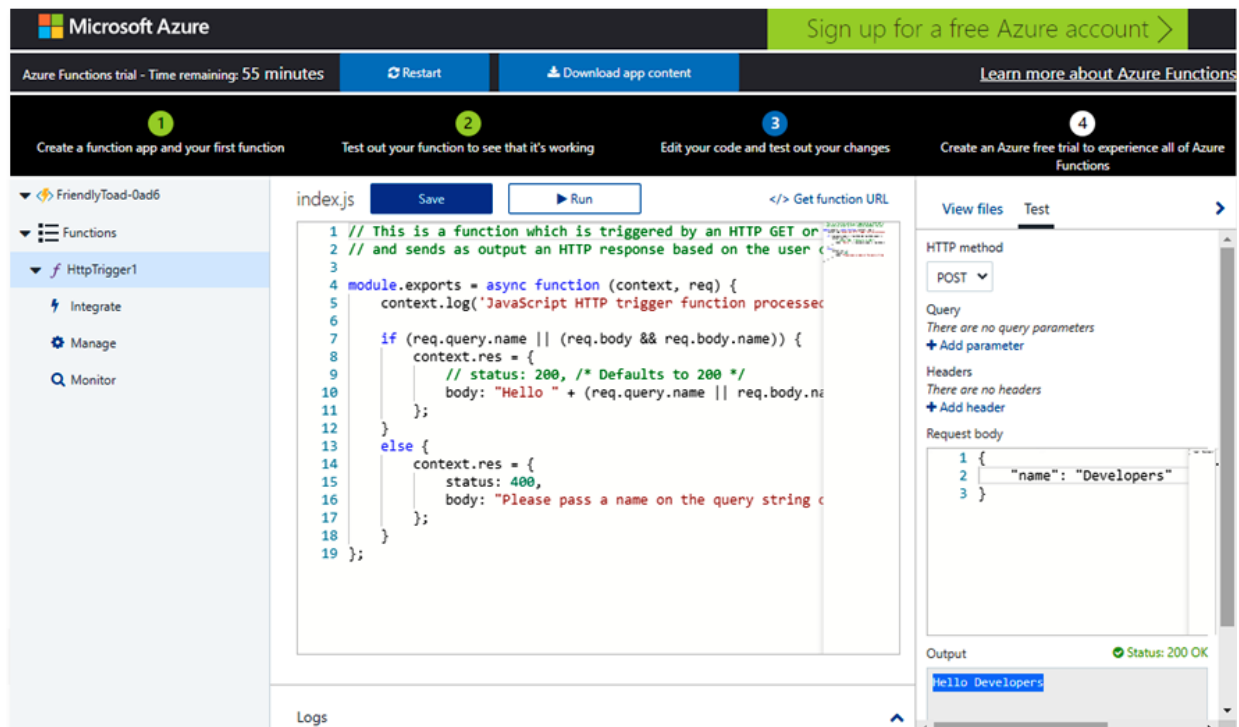


Abbildung 39: Überprüfen der Ausgabe nach der Ausführung

5. Ändern Sie nun die **HTTP method** (HTTP-Methode) in **GET**, und geben Sie den Abfragenamen und die Variable wie folgt ein. Betrachten Sie nach Klick auf die Schaltfläche **Run** (Ausführen) die Ausgabe:

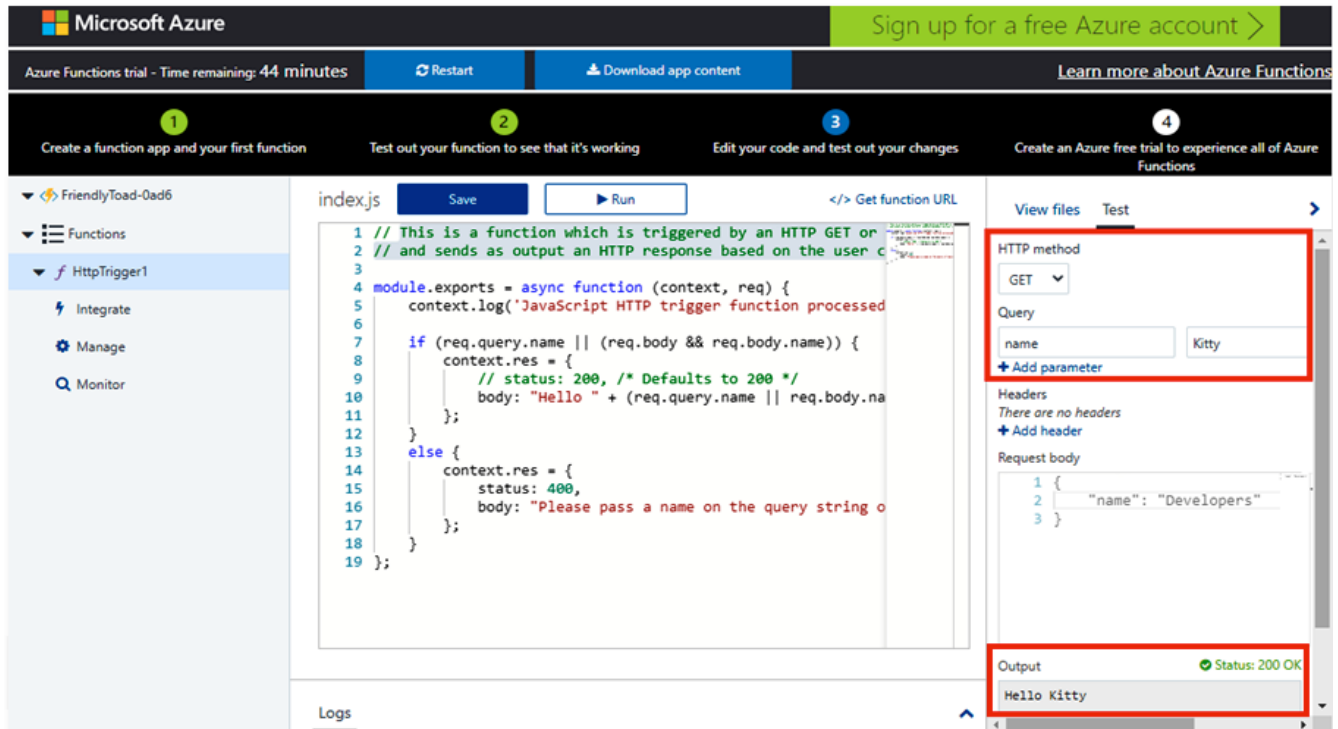


Abbildung 40: HTTP Get

6. Über diese kostenfreie Benutzeroberfläche können Sie auch weitere einfache Azure Functions-Proof-of-Concept-Apps entwickeln. Probieren Sie es aus und experimentieren Sie damit!

[Hier](#) erfahren Sie mehr über die Auswahl eines Azure-Computediensts für Ihre Anwendung. Im nächsten Abschnitt sind einige nützliche Lernmaterialien und Ressourcen aufgeführt.



Kapitel 5: Weiterführende Lernmaterialien und Ressourcen

Grundlagen von Azure erlernen

Als Neueinsteiger bei Azure empfehlen wir die folgenden interaktiven Lernpfade:

- [Azure-Grundlagen](#)
- [Cloudkonzepte von Microsoft Azure entdecken](#)
- [Unterscheidung zwischen den wichtigsten Microsoft Azure-Diensten](#)
- [Überprüfen der Sicherheit, des Datenschutzes, der Privatsphäre, der Compliance und der Vertrauensstellung von Microsoft Azure](#)
- [Überprüfen der Microsoft Azure-Preise, der Vereinbarungen zum Servicelevel und der Lebenszyklen](#)
- [Microsoft Learn](#)

Sobald Sie sich in Azure eingearbeitet haben, sollten Sie die Prüfung AZ-900 (Microsoft Azure-Grundlagen) ablegen, um sich zertifizieren zu lassen. Weitere Informationen finden Sie [hier](#).

Die für das Entwerfen eines Azure-Proof-of-Concept-Projekts nötigen Tools

In der folgenden Liste sind Tools aufgeführt, die für die in diesem Leitfaden gezeigten Beispiele unerlässlich sind:

- [Azure-Abonnement](#)
- [Visual Studio Code](#)
- [Azure Functions Core Tools](#)
- [GitHub-Konto](#)
- [Microsoft Edge-Browser \(basierend auf Chromium\)](#)
- [Node.js](#)

Weitere nützliche Ressourcen

Einführung in Azure Virtual Machines

- [Windows Virtual Machines-Dokumentation](#)
- [Linux Virtual Machines-Dokumentation](#)

Einführung in Azure Kubernetes Service

- [Azure Kubernetes Service](#)

Einführung in Azure App Service

- [App Service Migration-Tool](#)
- [Azure App Service: Übersicht](#)
- [Azure App Service testen](#)

Einführung in Azure Functions

- [Azure Functions-Dokumentation](#)
- [Konzepte für Azure Functions-Trigger und -Bindungen](#)
- [Azure Functions ausprobieren](#)

Kapitel 2: Beispielprojekt – Implementieren einer Web-App mit Azure Static Web Apps

- [Offizielle Dokumentation zu GitHub Actions](#)
- [Überprüfen von Pull Requests in Pre-Production-Umgebungen bei Azure Static Web Apps](#)

Kapitel 3: Beispielprojekt – Entwicklung eines intelligenten Chatbots

- [Microsoft Bot Framework SDK](#)
- [Beispiele für Bot Builder](#)
- [QnA Maker](#)

Allgemeine Ressourcen

- [Auswählen eines Azure-Computediensts für Ihre Anwendung](#)
- [Häufig gestellte Fragen zum kostenfreien Azure-Konto](#)



Schlussbemerkung

In diesem Leitfaden haben wir dargelegt, dass ein Proof-of-Concept-Projekt ein wertvolles Instrument darstellt, mit dem sich bewerten lässt, ob eine potenzielle Technologie oder ein Konzept die Anforderungen einer Geschäftslösung erfüllt. Damit können Sie mögliche technische und logistische Probleme vor der Implementierung in ein reguläres Projekt identifizieren. Ferner bietet es frühzeitige Insights in die Technologie und verringert Risiken, da wichtige Entscheidungen schon in der Anfangsphase des Entwicklungsprozesses getroffen werden können.

In „**Kapitel 1:** Proof-of-Concept-Leitfaden“ wurden die für die Planung und Ausführung eines erfolgreichen Proof-of-Concept-Projekts nötigen zentralen Grundlagen erläutert.

In „**Kapitel 2:** Beispielprojekt – Implementieren einer Web-App mit Azure Static Web Apps“ und „**Kapitel 3:** Beispielprojekt – Entwicklung eines intelligenten Chatbots“ wurden zwei praktische Projekte vorgestellt, die Sie als Inspiration für Ihre eigenen Proof-of-Concept-Projekte nutzen können.

In „**Kapitel 4:** Übersicht über Azure für Entwickler“ wurde die Azure-Plattform vorgestellt. Sie haben gelernt, wie Sie ein kostenfreies Azure-Konto anlegen, um direkt einzusteigen. Außerdem haben Sie erfahren, wie Sie das geeignete Cloudmodell und die passenden Dienste auswählen, um mit der Entwicklung Ihres Proof-of-Concept-Projekts für Azure zu beginnen.

In „**Kapitel 5:** Weiterführende Lernmaterialien und Ressourcen“ wurden zudem einige nützliche Lernmaterialien aufgeführt.

Sie sind nunmehr bereit, Ihr Proof-of-Concept-Projekt zu beginnen. Viel Glück!

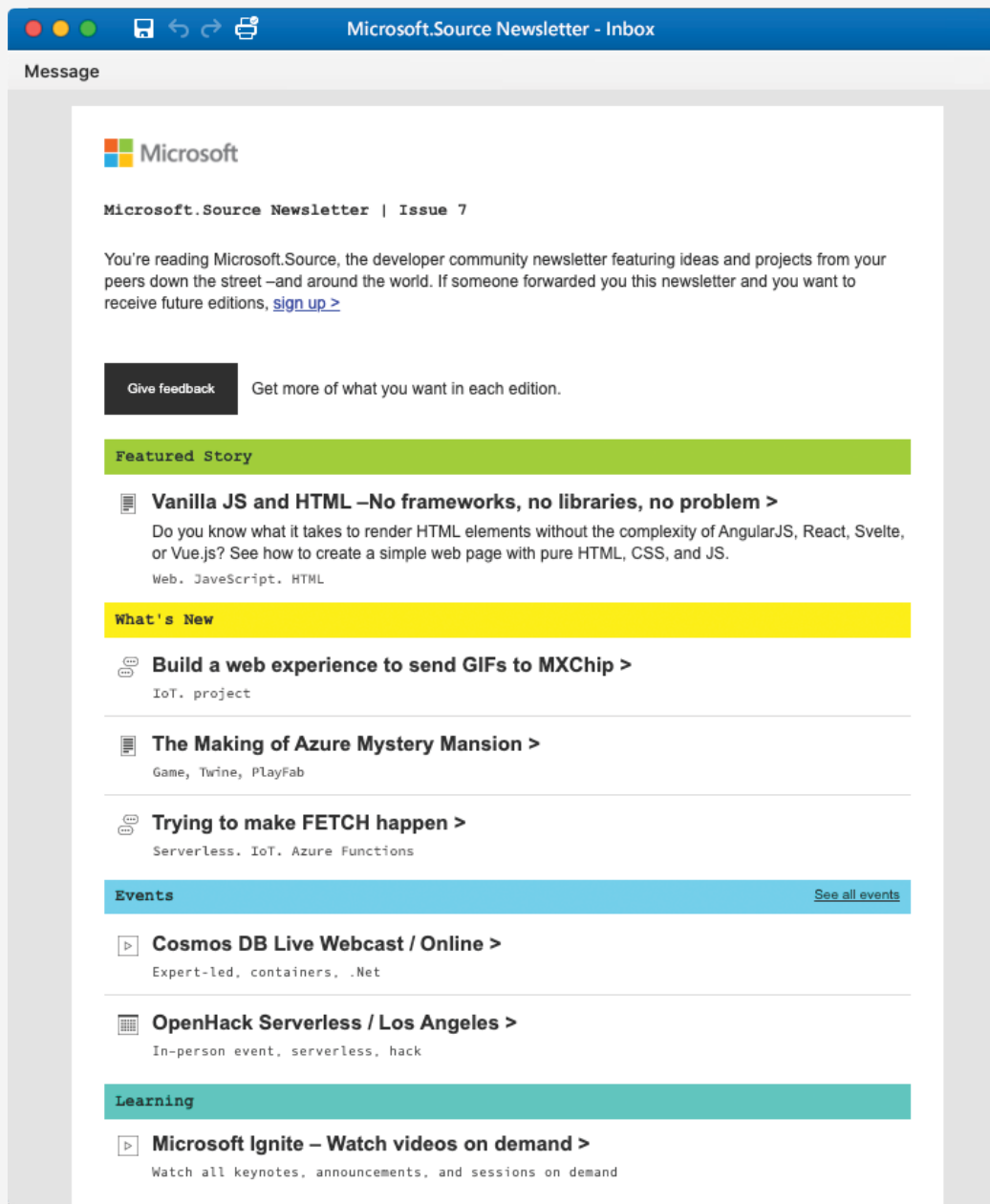


Starten Sie noch heute

 **Registrieren Sie sich für ein kostenfreies Azure-Konto.**

 **Erfahren Sie mehr über Azure-Lösungen.**

 **Sprechen Sie mit einem Vertriebsspezialisten, um mehr über Preise, Best Practices und die Umsetzung eines Proof-of-Concept-Projekts zu erfahren.**



Von Entwicklern für Entwickler

Microsoft.Source-Newsletter

In Microsoft.Source, dem sorgfältig zusammengestellten monatlichen Newsletter der Entwicklercommunity, erhalten Sie technische Artikel, Beispielcode und Informationen zu bevorstehenden Veranstaltungen.

- Klären Sie sich über die neuesten Technologien
- Vernetzen Sie sich mit Ihren Kollegen bei Community-Events
- Lernen Sie mit praktischen Ressourcen



Registrieren