

Einführung

1 Minute

100 XP

In diesem Modul werden die Funktionen und Tools für die Verwaltung und Bereitstellung von Azure-Ressourcen vorgestellt. Sie erfahren mehr über das Azure-Portal (eine grafische Benutzeroberfläche für die Verwaltung von Azure-Ressourcen), über die Befehlszeile und Skripttools, die beim Bereitstellen oder Konfigurieren von Ressourcen hilfreich sind. Darüber hinaus erfahren Sie mehr über Azure-Dienste, die Ihnen dabei helfen, Ihre lokalen und Multicloud-Umgebungen in Azure zu verwalten.

Lernziele

Nach Abschluss dieses Moduls können Sie folgende Aufgaben durchführen:

- Beschreiben Sie das Azure-Portal.
- Beschreiben von Azure Cloud Shell, einschließlich der Azure CLI und Azure PowerShell
- Beschreiben des Zwecks von Azure Arc
- Beschreiben Sie Azure Resource Manager (ARM), ARM-Vorlagen und Bicep.

Nächste Lektion: Beschreiben von Tools für die Interaktion mit Azure

Beschreiben von Tools für die Interaktion mit Azure

100 XP

5 Minuten

Um Azure optimal nutzen zu können, benötigen Sie eine Möglichkeit zur Interaktion mit der Azure-Umgebung, den Verwaltungsgruppen, Abonnements, Ressourcengruppen, Ressourcen usw. Azure bietet mehrere Tools zum Verwalten Ihrer Umgebung:

- Azure-Portal
- Azure PowerShell
- Azure-Befehlszeilenschnittstelle (CLI)

Was ist das Azure-Portal?

Das Azure-Portal ist eine webbasierte, zentrale Konsole, die eine Alternative zu Befehlszeilentools darstellt. Mit dem Azure-Portal können Sie Ihr Azure-Abonnement über eine grafische Benutzeroberfläche verwalten. Ihre Möglichkeiten:

- Erstellen, Verwalten und Überwachen aller Aspekte – von einfachen Web-Apps bis hin zu komplexen Cloudbereitstellungen
- Erstellen benutzerdefinierter Dashboards für eine übersichtliche Ansicht der Ressourcen
- Konfigurieren von Optionen für die Barrierefreiheit für eine optimale Benutzererfahrung

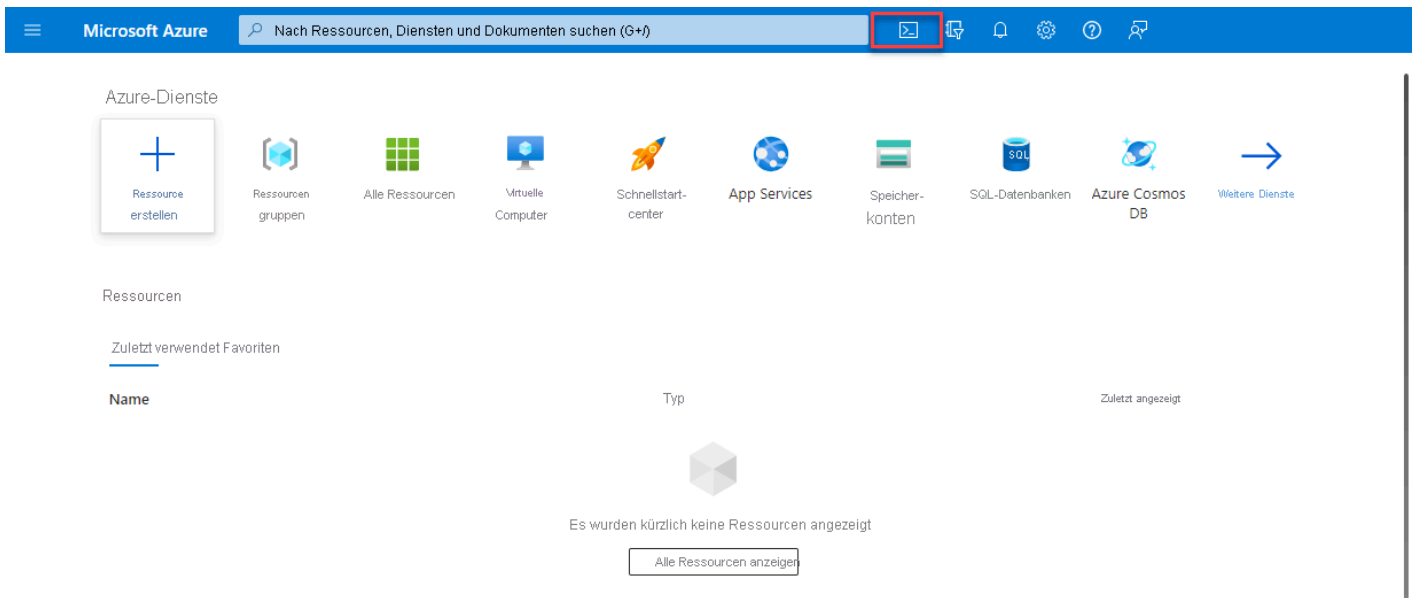
Im folgenden Video finden Sie eine Einführung in das Azure-Portal:

Das Azure-Portal ist auf Resilienz und fortlaufende Verfügbarkeit ausgelegt. Es ist in jedem Azure-Rechenzentrum vorhanden. Diese Konfiguration sorgt dafür, dass das Azure-Portal Ausfällen einzelner Rechenzentren gegenüber resilient ist. Darüber hinaus werden durch die Nähe zu den Benutzern Verlangsamungen des Netzwerks vermieden. Das Azure-Portal wird ständig aktualisiert und erfordert bei Wartungsarbeiten keine Downtime.

Azure Cloud Shell

Azure Cloud Shell ist ein browserbasiertes Shelltool, mit dem Sie Azure-Ressourcen mithilfe einer Shell erstellen, konfigurieren und verwalten können. Azure Cloud Shell unterstützt sowohl Azure PowerShell als auch die Azure-Befehlszeilenschnittstelle (CLI), die eine Bash-Shell ist.

Sie können über das Azure-Portal auf Azure Cloud Shell zugreifen, indem Sie das Cloud Shell-Symbol auswählen:



Azure Cloud Shell bietet mehrere Features, die es zu einem einzigartigen Angebot für die Verwaltung von Azure machen. Dazu gehören unter anderem die folgenden Features:

- Es handelt sich um eine browserbasierte Shell, die ohne lokale Installation oder Konfiguration auskommt.
- Die Authentifizierung erfolgt mit Ihren Azure-Anmeldeinformationen. Wenn Sie sich anmelden, weiß sie daher sofort, wer Sie sind und welche Berechtigungen Sie haben.
- Sie können die Shell auswählen, mit der Sie am meisten vertraut sind. Azure Cloud Shell unterstützt sowohl Azure PowerShell als auch die Azure-Befehlszeilenschnittstelle (die Bash verwendet).

Was ist Azure PowerShell?

Azure PowerShell ist eine Shell, mit der Entwickler*innen sowie DevOps- und IT-Expert*innen Befehle ausführen können, die als Cmdlets (ausgesprochen „Command-lets“) bezeichnet werden. Mit diesen Befehlen wird die Azure-REST-API aufgerufen, um Verwaltungsaufgaben in Azure auszuführen. Cmdlets können unabhängig ausgeführt werden, um einmalige Änderungen vorzunehmen, oder sie können kombiniert werden, um komplexe Aktionen wie die folgenden zu orchestrieren:

- Die routinemäßige Einrichtung, Bereinigung und Verwaltung einer einzelnen Ressource oder mehrerer verbundener Ressourcen.
- Die Bereitstellung einer vollständigen Infrastruktur, die Dutzende oder Hunderte von Ressourcen enthalten kann, mit imperativem Code.

Die Erfassung der Befehle in einem Skript macht den Prozess wiederholbar und automatisierbar.

Zusätzlich zur Verfügbarkeit über Azure Cloud Shell können Sie Azure PowerShell auf Windows-, Linux- und Mac-Plattformen installieren und konfigurieren.

Was ist die Azure CLI?

Die Azure-Befehlszeilenschnittstelle entspricht funktionell Azure PowerShell. Der Hauptunterschied liegt in der Syntax von Befehlen. Während bei Azure PowerShell PowerShell-Befehle verwendet werden, sind es bei der Azure-Befehlszeilenschnittstelle Bash-Befehle.

Die Azure-Befehlszeilenschnittstelle bietet dieselben Vorteile bei der Durchführung von diskreten Aufgaben oder dem Orchestrieren komplexer Vorgänge per Code. Sie kann ebenfalls auf Windows-, Linux- und Mac-Plattformen sowie über Azure Cloud Shell installiert werden.

Aufgrund der Ähnlichkeiten bei den Funktionen und dem Zugriff zwischen Azure PowerShell und der Bash-basierten Azure-Befehlszeilenschnittstelle kommt es für Sie hauptsächlich auf die Sprache an, mit der Sie vertraut sind.

Nächste Lektion: Beschreiben des Zwecks von Azure Arc

Beschreiben des Zwecks von Azure Arc

100 XP

3 Minuten

Das Verwalten von Hybrid- und Multi-Cloud-Umgebungen kann schnell kompliziert werden. Azure bietet eine Vielzahl von Tools zum Bereitstellen, Konfigurieren und Überwachen von Azure-Ressourcen. Was ist aber mit den lokalen Ressourcen in einer Hybridkonfiguration oder den Cloudressourcen in einer Multi-Cloud-Konfiguration?

Mithilfe von Azure Resource Manager (ARM) und Arc können Sie Ihre Azure-Compliance und -Überwachung auf Ihre Hybrid- und Multi-Cloud-Konfigurationen ausdehnen. Azure Arc bietet eine konsistente Verwaltungsplattform für mehrere Clouds und die lokale Umgebung und vereinfacht so die Governance und Verwaltung.

Azure Arc bietet eine zentralisierte, einheitliche Möglichkeit für Folgendes:

- Sie verwalten Ihre gesamte Umgebung zusammen, indem Sie Ihre vorhandenen Azure-fremden Ressourcen in ARM projizieren.
- Sie verwalten Multi-Cloud- und Hybrid-VMs, Kubernetes-Cluster und Datenbanken so, als würden sie in Azure ausgeführt.
- Sie können ortsunabhängig vertraute Azure-Dienste und Verwaltungsfunktionen verwenden.
- Sie können herkömmliche ITOps-Methoden weiterverwenden und gleichzeitig DevOps-Methoden einführen, um neue cloudnative Muster in Ihrer Umgebung zu unterstützen.
- Konfigurieren Sie benutzerdefinierte Speicherorte als Abstraktionsschicht auf der Grundlage von Kubernetes-Clustern mit Azure Arc-Unterstützung und Clustererweiterungen.

Wofür können Sie Azure Arc außerhalb von Azure nutzen?

Aktuell können mit Azure Arc folgende außerhalb von Azure gehostete Ressourcentypen verwaltet werden:

- Server
- Kubernetes-Cluster
- Azure-Datendienste
- SQL Server
- VMs (Vorschau)

Nächste Lektion: Beschreiben von Azure Resource Manager und Azure ARM-Vorlagen

Beschreiben von Azure Resource Manager und Azure ARM-Vorlagen

100 XP

8 Minuten

Azure Resource Manager (ARM) ist der Bereitstellungs- und Verwaltungsdienst für Azure. Er bietet eine Verwaltungsebene, die das Erstellen, Aktualisieren und Löschen von Ressourcen in Ihrem Azure-Konto ermöglicht. Was auch immer Sie mit Ihren Azure-Ressourcen erledigen, ARM ist daran beteiligt.

Wenn Benutzer*innen eine Anforderung über eines der Azure-Tools, eine der APIs oder eines der SDKs senden, wird diese von ARM empfangen. ARM authentifiziert und autorisiert die Anforderung. Anschließend sendet ARM die Anforderung an den Azure-Dienst, der daraufhin die angeforderte Aktion ausführt. Da alle Anforderungen von derselben API verarbeitet werden, stehen in allen Tools konsistente Ergebnisse und Funktionen zur Verfügung.

Vorteile von Azure Resource Manager

Mit Azure Resource Manager können Sie folgende Aktionen ausführen:

- Verwalten Ihrer Infrastruktur über deklarative Vorlagen (anstelle von Skripts) Eine Resource Manager-Vorlage ist eine JSON-Datei, mit der definiert wird, was Sie in Azure bereitstellen möchten.
- Bereitstellen, Verwalten und Überwachen aller Ressourcen für Ihre Lösung als Gruppe, anstatt diese Ressourcen einzeln zu behandeln
- Sie können Ihre Lösung während des Entwicklungszyklus erneut bereitstellen und sicher sein, dass Ihre Ressourcen in einem konsistenten Zustand bereitgestellt werden.
- Sie können die Abhängigkeiten zwischen Ressourcen definieren, damit diese in der richtigen Reihenfolge bereitgestellt werden.
- Wenden Sie die Zugriffssteuerung auf alle Dienste an, da RBAC nativ in die Verwaltungsplattform integriert ist.
- Anwenden von Tags auf Ressourcen, um alle Ressourcen in Ihrem Abonnement logisch zu organisieren
- Sie können genauere Informationen zur Abrechnung Ihrer Organisation erhalten, indem Sie die Kosten für eine Gruppe von Ressourcen anzeigen, die über dasselbe Tag verfügen.

Im folgenden Video erhalten Sie eine Übersicht zum Azure Resource Manager.

Infrastructure-as-Code

Infrastructure-as-Code ist ein Konzept, bei dem Sie Ihre Infrastruktur als Codezeilen verwalten. Auf einführender Ebene geht es um die Verwendung von Azure Cloud Shell, Azure PowerShell oder der Azure CLI zum Verwalten und Konfigurieren Ihrer Ressourcen. Mit zunehmender Vertrautheit mit der Cloud können Sie das Infrastructure-as-Code-Konzept verwenden, um ganze Bereitstellungen mit wiederholbaren Vorlagen und Konfigurationen zu verwalten. ARM-Vorlagen und Bicep sind zwei Beispiele für die Verwendung von Infrastructure-as-Code mit dem Azure-Resource Manager, um Ihre Umgebung zu verwalten.

ARM-Vorlagen

Mit ARM-Vorlagen können Sie die Ressourcen, die Sie verwenden möchten, in einem deklarativen JSON-Format beschreiben. Bei einer ARM-Vorlage wird der Bereitstellungscode überprüft, bevor

Code ausgeführt wird. Dadurch wird sichergestellt, dass die Ressourcen ordnungsgemäß erstellt und verbunden werden. Die Vorlage orchestriert dann die Erstellung dieser Ressourcen parallel. Das bedeutet: Wenn Sie 50 Instanzen derselben Ressource benötigen, werden alle 50 Instanzen gleichzeitig erstellt.

Letztlich muss der Entwickler, DevOps- oder IT-Experte nur den gewünschten Zustand und die gewünschte Konfiguration jeder Ressource in der ARM-Vorlage definieren, und die Vorlage erledigt den Rest. Über Vorlagen können sogar PowerShell- und Bash-Skripts vor oder nach dem Einrichten einer Ressource ausgeführt werden.

Vorteile der Verwendung von ARM-Vorlagen

ARM-Vorlagen bieten viele Vorteile beim Planen der Bereitstellung von Azure-Ressourcen. Zu den wichtigsten Vorteilen gehören folgende:

- **Deklarative Syntax:** Mit ARM-Vorlagen können Sie eine gesamte Azure-Infrastruktur deklarativ erstellen und bereitstellen. Deklarative Syntax bedeutet, dass Sie deklarieren, was Sie bereitstellen möchten, aber nicht die tatsächlichen Programmierbefehle und die Sequenz für die Ressourcenbereitstellung schreiben müssen.
- **Wiederholbare Ergebnisse:** Wiederholtes Bereitstellen Ihrer Infrastruktur während des gesamten Entwicklungslebenszyklus und Gewährleistung, dass Ihre Ressourcen einheitlich bereitgestellt werden. Sie können dieselbe ARM-Vorlage verwenden, um mehrere Dev/Test-Umgebungen bereitzustellen, und können dabei sicher sein, dass alle Umgebungen identisch sind.
- **Orchestrierung:** Sie müssen sich keine Gedanken über die Komplexität von Bestellvorgängen machen. Azure Resource Manager orchestriert die Bereitstellung unabhängiger Ressourcen, sodass sie in der richtigen Reihenfolge erstellt werden. Wenn möglich, stellt Azure Resource Manager Ressourcen parallel bereit, sodass Ihre Bereitstellungen schneller abgeschlossen werden als serielle Bereitstellungen. Sie stellen die Vorlage über einen Befehl statt über mehrere imperativer Befehle bereit.
- **Modulare Dateien:** Sie können Ihre Vorlagen in kleinere, wiederverwendbare Komponenten unterteilen und sie zum Zeitpunkt der Bereitstellung miteinander verknüpfen. Sie können eine Vorlage auch in anderen Vorlagen schachteln. Sie können beispielsweise eine Vorlage für einen VM-Stapel erstellen und dann diese Vorlage in Vorlagen schachteln, mit denen eine vollständige Umgebungen bereitgestellt wird. Dieser VM-Stapel wird dann in jeder der Umgebungsvorlagen konsistent bereitgestellt.
- **Erweiterbarkeit:** Mit Bereitstellungsskripts können Sie Ihren Vorlagen PowerShell- oder Bash-Skripts hinzufügen. Die Bereitstellungsskripts erweitern die Möglichkeiten zum Einrichten von Ressourcen während der Bereitstellung. Ein Skript kann in die Vorlage eingeschlossen werden oder in einer externen Quelle gespeichert werden, um dann in der Vorlage darauf zu verweisen. Mit Bereitstellungsskripts haben Sie die Möglichkeit, die Einrichtung der End-to-End-Umgebung in einer einzelnen ARM-Vorlage abzuschließen.

Bicep

Bicep ist eine Sprache, die eine deklarative Syntax zur Bereitstellung von Azure-Ressourcen verwendet. Eine Bicep-Datei definiert die Infrastruktur und Konfiguration. Anschließend stellt ARM diese Umgebung basierend auf Ihrer Bicep-Datei bereit. Bicep-Dateien ähneln zwar einer ARM-Vorlage, die in JSON geschrieben ist, haben tendenziell aber einen einfacheren, prägnanteren Stil.

Einige Vorteile von Bicep sind:

- **Unterstützung für alle Ressourcentypen und API-Versionen:** In Bicep werden alle Vorschau- und allgemein verfügbaren Versionen für Azure-Dienste direkt unterstützt. Sobald ein Ressourcenanbieter neue Ressourcentypen und API-Versionen einführt, können Sie diese

in der Bicep-Datei verwenden. Sie müssen nicht warten, bis Tools aktualisiert werden, um die neuen Dienste verwenden zu können.

- **Einfache Syntax:** Im Vergleich zur entsprechenden JSON-Vorlage sind Bicep-Dateien präziser und einfacher zu lesen. Für Bicep sind keine Vorkenntnisse über Programmiersprachen erforderlich. Die Bicep-Syntax ist deklarativ und gibt an, welche Ressourcen und Ressourceneigenschaften Sie bereitstellen möchten.
- **Wiederholbare Ergebnisse:** Wiederholtes Bereitstellen Ihrer Infrastruktur während des gesamten Entwicklungslebenszyklus und Gewährleistung, dass Ihre Ressourcen einheitlich bereitgestellt werden. Bicep-Dateien sind idempotent, d. h. Sie können dieselbe Datei mehrmals bereitstellen und erhalten immer dieselben Ressourcentypen im selben Zustand. Sie können eine Datei entwickeln, die den gewünschten Zustand darstellt, anstatt viele separate Dateien zur Darstellung von Aktualisierungen zu entwickeln.
- **Orchestrierung:** Sie müssen sich keine Gedanken über die Komplexität von Bestellvorgängen machen. Resource Manager orchestriert die Bereitstellung voneinander abhängiger Ressourcen, sodass diese in der richtigen Reihenfolge erstellt werden. Wenn möglich, stellt der Resource Manager Ressourcen parallel bereit, sodass Ihre Bereitstellungen schneller abgeschlossen werden als serielle Bereitstellungen. Sie stellen die Datei über einen Befehl statt über mehrere imperativer Befehle bereit.
- **Modularität:** Sie können den Bicep-Code mithilfe von Modulen in verwaltbare Teile unterteilen. Über ein Modul werden zugehörige Ressourcen bereitgestellt. Mit Modulen lässt sich Code wiederverwenden und die Entwicklung vereinfachen. Sie können ein Modul jederzeit in einer Bicep-Datei hinzufügen, wenn Sie die entsprechenden Ressourcen bereitstellen möchten.

Nächste Lektion: Wissensbeurteilung