



Auswahl einer Infrastruktur als Codetool für Ihr Unternehmen

AWS Prescriptive Guidance



AWS Prescriptive Guidance: Auswahl einer Infrastruktur als Codetool für Ihr Unternehmen

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Die Handelsmarken und die Handelsaufmachung von Amazon dürfen nicht in einer Weise in Verbindung mit nicht von Amazon stammenden Produkten oder Services verwendet werden, durch die Kunden irregeführt werden könnten oder Amazon in schlechtem Licht dargestellt oder diskreditiert werden könnte. Alle anderen Handelsmarken, die nicht Eigentum von Amazon sind, gehören den jeweiligen Besitzern, die möglicherweise zu Amazon gehören oder nicht, mit Amazon verbunden sind oder von Amazon gesponsert werden.

Table of Contents

Einführung	1
Vorteile von IaC	2
IaC-Tools	4
CloudFormation	4
AWS SAM	6
AWS CDK	7
Terraform	8
Pulumi	10
Ein Tool auswählen	12
Nächste Schritte	13
Ressourcen	14
AWS Dokumentation	14
Weitere Dokumentation	14
Tools	14
Mitwirkende	15
Inhaltserstellung	15
Überprüfung	15
Technisches Schreiben	15
Dokumentverlauf	16
.....	xvii

Auswahl einer Infrastruktur als Codetool für Ihr Unternehmen

Amazon Web Services ([Mitwirkende](#))

Februar 2026 ([Verlauf der Dokumente](#))

Infrastructure as Code (IaC) ist der Prozess der Bereitstellung und Verwaltung der Infrastruktur einer Anwendung mithilfe einer Reihe von Konfigurationsdateien. IaC soll Ihnen helfen, das Infrastrukturmanagement zu zentralisieren, Ressourcen zu standardisieren und schnell zu skalieren, sodass neue Umgebungen wiederholbar, zuverlässig und konsistent sind. Es ist eine Schlüsselkomponente von Agile und DevOps Praktiken wie Versionskontrolle, kontinuierliche Integration und kontinuierliche Bereitstellung.

Die Wahl eines Tools für Infrastructure as Code (IaC) wird als strategische Entscheidung für ein Unternehmen angesehen. Diese Entscheidung betrifft alle Teams, die Infrastruktur, Anwendungen und Dienste für das Unternehmen entwickeln. Jedes Tool hat Vor- und Nachteile. Daher gibt es kein one-size-fits-all Modell.

In der Vergangenheit war die Verwaltung und Bereitstellung der Infrastruktur ein manueller Prozess mit vielen Fehlern. IaC optimiert diese Aufgaben mithilfe von Code und hat sich zu einer zuverlässigen Lösung für die Bereitstellung von Infrastruktur entwickelt. IaC-Tools ermöglichen es Entwicklern, die Infrastruktur mithilfe von Programmiersprachen zu definieren und bereitzustellen. Dies erhöht nicht nur die Flexibilität des Unternehmens, sondern beschleunigt auch das Wachstum und die Innovationsgeschwindigkeit. Darüber hinaus verbessert IaC die Sicherheit erheblich, da IaC es Ihrem Unternehmen ermöglicht, den Code vor der Bereitstellung zu scannen und so zu überprüfen, ob die Infrastruktur zuverlässig und sicher ist. Letztlich ist das richtige IaC-Tool nicht nur eine technische, sondern eine strategische Entscheidung, die sich direkt auf den Gesamterfolg des Unternehmens auswirkt.

In diesem Leitfaden werden fünf verschiedene IaC-Tools vorgestellt, die zur Bereitstellung von AWS Ressourcen verwendet werden können: AWS CloudFormation, AWS Serverless Application Model (AWS SAM) AWS Cloud Development Kit (AWS CDK), HashiCorp Terraform und Pulumi. Es vergleicht diese Tools und führt Sie durch den Prozess der Auswahl eines Tools, das den Anforderungen Ihres Teams, Ihrer Organisation und Ihres Cloud-Talents entspricht. Der Schlüssel liegt darin, das gewählte IaC-Tool auf Ihre Unternehmensziele und die Fähigkeiten Ihrer Entwickler abzustimmen. Wenn Ihr Team sich beispielsweise auskennt JavaScript, können Sie es TypeScript als primäres IaC-Tool wählen AWS CDK , da es Ihren Entwicklungsablauf optimiert.

Vorteile der Implementierung von IaC

Durch die Auswahl des richtigen IaC-Tools für Ihr Unternehmen können Sie die folgenden Vorteile erzielen:

- **Geschwindigkeit** — Ein Ziel von IaC ist es, die Dinge zu beschleunigen, indem manuelle Prozesse vermieden werden. Ein codebasierter Ansatz macht es einfacher, in kürzerer Zeit mehr zu erledigen. Die manuelle Einrichtung jeder IT-Umgebung ist sehr teuer und erfordert engagierte Techniker und Architekten für die Einrichtung der Hard- und Software. Durch die Erhöhung der Geschwindigkeit sind Unternehmen besser in der Lage, sich schnell an sich ändernde Kundenbedürfnisse und Marktbedingungen anzupassen. Außerdem können Sie Ihren IaC-Code einmal schreiben und denselben Code in Hunderten von Umgebungen bereitstellen, und das in einem Bruchteil der Zeit, die Sie für die manuelle Erstellung benötigen würden.
- **Skalierbarkeit** — IaC erleichtert das Hinzufügen von Ressourcen zur vorhandenen Infrastruktur. Upgrades werden schnell bereitgestellt, sodass Sie in Spitzenzeiten schnell expandieren können. Unternehmen, die Onlinedienste anbieten, können beispielsweise skalieren, um den Benutzeranforderungen in Zeiten hoher Nachfrage, wie z. B. am Black Friday, gerecht zu werden.
- **Verbesserte Sicherheit** — IaC zeichnet sich dadurch aus, dass es Unternehmen Sicherheit und Compliance bietet. Organizations können grundlegende Sicherheitsrichtlinien und -konfigurationen festlegen und diese über Pipelines durchsetzen, indem sie automatisierte Tests anhand Ihrer IaC durchführen. Wenn die IaC gegen Compliance-Regeln verstößt, schlägt die Pipeline fehl und gibt dem Entwickler automatisch Feedback. Dies kann die Schaffung einer unsicheren Infrastruktur verhindern.
- **Konsistenz** — Konsistenz ist ein weiterer wichtiger Vorteil von IaC. Wenn mehrere Techniker Konfigurationen manuell bereitstellen, sind Inkonsistenzen unvermeidlich. Mit der Zeit wird es schwierig, dieselben Umgebungen nachzuverfolgen und zu reproduzieren. Diese Inkonsistenzen führen häufig zu gravierenden Unterschieden zwischen Entwicklungs-, QA- und Produktionsumgebungen.
- **Effizienz** — IaC verbessert die Effizienz und Produktivität während des gesamten Entwicklungszyklus. Programmierer erstellen Sandbox-Umgebungen, um sie isoliert zu entwickeln. Der Betrieb kann schnell die Infrastruktur für Sicherheitstests bereitstellen. Die QA-Techniker verfügen während der Tests über exakte Kopien der Produktionsumgebungen. Wenn sie bereit für die Bereitstellung sind, stellen Entwickler sowohl die Infrastruktur als auch den Code in einem Schritt zur Produktion bereit. Es kann auch ein höheres Maß an Zusammenarbeit und Teamwork

ermöglichen. Teams können sichere Muster für Anwendungen erstellen und diese Vorlagen oder Konstrukte dann mit anderen Teams teilen, wodurch Doppelarbeit reduziert wird.

- **Geringere Kosten** — IaC reduziert die Kosten für die Softwareentwicklung. Es besteht keine Notwendigkeit, Ressourcen für die manuelle Einrichtung von Umgebungen aufzuwenden. Sie zahlen nur für die Ressourcen, die Sie aktiv nutzen, sodass kein unnötiger Mehraufwand entsteht.
- **Verbesserte Zuverlässigkeit** — Wenn Ihre Infrastruktur groß ist, kann es leicht sein, eine Ressource falsch zu konfigurieren oder Dienste in der falschen Reihenfolge bereitzustellen. Mit IaC werden die Ressourcen immer genau wie deklariert bereitgestellt und konfiguriert. Da das manuelle Erstellen von Ressourcen fehleranfällig ist, können Sie sich darauf verlassen, dass Ihre Infrastruktur wie vorgesehen erstellt wird, wenn Sie IaC verwenden.
- **Erkennung von Konfigurationsabweichungen** — Konfigurationsabweichungen treten auf, wenn die Konfiguration, mit der Ihre Umgebung bereitgestellt wurde, nicht mehr mit der tatsächlichen Umgebung übereinstimmt. Viele IaC-Tools helfen Ihnen dabei, Abweichungen zu erkennen und zu beheben. Wenn beispielsweise jemand Ihre Ressourcen fälschlicherweise manuell ändert, können Sie das IaC-Tool verwenden, um festzustellen, welche Änderungen vorgenommen wurden, und sie in den vorgesehenen Zustand zurückversetzen.
- **Experimentieren** — Da IaC die Bereitstellung neuer Infrastrukturen erheblich schneller und einfacher macht, können Sie experimentelle Änderungen vornehmen und testen, ohne viel Zeit und Ressourcen investieren zu müssen. Wenn Ihnen die Ergebnisse gefallen, können Sie die neue Infrastruktur schnell für die Produktion skalieren.

Überprüfung der IaC-Tools für AWS Cloud

In diesem Handbuch werden fünf verschiedene IaC-Tools vorgestellt, die zur Bereitstellung AWS von Ressourcen verwendet werden können. Es vergleicht diese Tools und führt Sie durch den Prozess der Auswahl eines Tools, das den Anforderungen Ihres Teams, Ihrer Organisation und Ihres Cloud-Talents entspricht. Der Schlüssel liegt darin, das gewählte IaC-Tool auf Ihre Unternehmensziele und die Fähigkeiten Ihrer Entwickler abzustimmen.

In diesem Abschnitt erfahren Sie mehr darüber, wie die einzelnen Tools funktionieren und welche Vor- und Nachteile sie haben. Einige Tools können beispielsweise nur in der verwendet werden AWS Cloud, und sie eignen sich nicht gut für Multi-Cloud- oder Hybrid-Cloud-Bereitstellungen. Andere erfordern Kenntnisse in speziellen Programmiersprachen, und andere unterstützen gängige Programmiersprachen. Bewerten Sie die Vor- und Nachteile der einzelnen Tools und überlegen Sie, welches für Ihr Unternehmen am besten geeignet ist.

In diesem Abschnitt werden die folgenden IaC-Tools beschrieben:

- [AWS CloudFormation](#)
- [AWS Serverless Application Model \(AWS SAM\)](#)
- [AWS Cloud Development Kit \(AWS CDK\)](#)
- [HashiCorp Terraform](#)
- [Pulumi](#)

Verwendung AWS CloudFormation als IaC-Tool

[AWS CloudFormation](#) ist ein Programm AWS-Service, das Vorlagendateien verwendet, um die Bereitstellung von AWS Ressourcen zu automatisieren. Sie erstellen eine Vorlage, die alle AWS Ressourcen beschreibt, die Sie bereitstellen möchten, und diese Ressourcen für Sie CloudFormation bereitstellt und konfiguriert.

CloudFormation Vorlagen werden mithilfe von JSON oder YAML geschrieben. Ein CloudFormation Stack ist die Implementierung der in Ihrer Vorlage definierten Ressourcen. Sie können Ihre CloudFormation Stacks über das AWS Management Console, programmgesteuert über das CloudFormation SDK oder über () verwalten. AWS Command Line Interface AWS CLI Weitere Informationen zur CloudFormation Funktionsweise finden Sie unter [AWS CloudFormation Konzepte](#) und [AWS CloudFormation Funktionsweise](#) in der Dokumentation. CloudFormation

Vorteile der Verwendung von CloudFormation:

- CloudFormation Mit [Änderungssätzen](#) können Sie eine Vorschau der Änderungen an einem laufenden Stack anzeigen, bevor Sie diese Änderungen bereitstellen. Änderungssätze fassen die vorgeschlagenen Änderungen an den laufenden Ressourcen in einem vorhandenen Stack zusammen. Dies kann Ihnen helfen, Konflikte oder unbeabsichtigte Folgen vor der Bereitstellung zu identifizieren. Wenn Sie beispielsweise den Namen einer Amazon Relational Database Service (Amazon RDS) -Datenbank-Instance ändern, CloudFormation wird eine neue Datenbank erstellt und die alte gelöscht. Sie würden die Daten in der alten Datenbank verlieren, sofern Sie sie nicht bereits gesichert haben. Wenn Sie einen Änderungssatz generieren, stellen Sie fest, dass Ihre Änderung dazu führt, dass Ihre Datenbank ersetzt wird, und Sie können entsprechend planen, bevor Sie Ihren Stack aktualisieren.
- Wenn bei der Bereitstellung eines Änderungssatzes ein Fehler auftritt, CloudFormation wird automatisch zum letzten bekannten Betriebsstatus zurückgesetzt.
- Sie können CloudFormation [Stacksets](#) verwenden, um Ressourcen für mehrere AWS-Konten und bereitzustellen AWS-Regionen.
- Für die Verwendung CloudFormation mit Ressourcenanbietern in den folgenden Namespaces fallen keine zusätzlichen Gebühren an: AWS: :*, Alexa: :* und Custom: :*. In diesen Fällen zahlen Sie nur für die AWS Ressourcen, die Sie bereitstellen, als ob Sie sie manuell bereitgestellt hätten.
- CloudFormation verwaltet den Status für Sie. Das bedeutet, dass CloudFormation die zugrunde liegenden Dienste aufgerufen werden AWS , um Ihre Ressourcen gemäß den Definitionen in Ihren CloudFormation Vorlagen bereitzustellen und zu konfigurieren.
- CloudFormation bietet Tools zur Erkennung und Behebung von Konfigurationsabweichungen. Weitere Informationen finden Sie in der [Dokumentation unter Erkennen nicht verwalteter Konfigurationsänderungen an Stacks und Ressourcen](#). CloudFormation
- Sie können es verwenden CloudFormation , um [benutzerdefinierte](#) Ressourcen zu erstellen. Sie können benutzerdefinierte Bereitstellungslogik in Vorlagen schreiben, die jedes Mal CloudFormation ausgeführt wird, wenn Sie Stacks erstellen, aktualisieren oder löschen.
- CloudFormation [unterstützt die Modellierung, Bereitstellung und Verwaltung von Anwendungsressourcen von Drittanbietern mit Registrierung](#). CloudFormation
- CloudFormation unterstützt [den Import vorhandener Ressourcen](#) in die CloudFormation Verwaltung.

Nachteile der Verwendung von CloudFormation:

- Wenn Sie mit der JSON- oder YAML-Syntax nicht vertraut sind, kann es etwas gewöhnungsbedürftig sein. JSON wurde nicht so konzipiert, dass es für Menschen lesbar ist, und es erlaubt Ihnen auch nicht, Inline-Kommentare abzugeben. YAML ermöglicht das Verfassen von Kommentaren und ist einfacher zu lesen. Die Syntax basiert jedoch auf Tabulatoren und Leerzeichen, sodass es leicht sein kann, Fehler bei Einrückungen zu machen.
- CloudFormation unterstützt keine Multi-Cloud-Bereitstellungen.
- Sie müssen eine Implementierung auf höherer Ebene verwenden, z. B. die, um wiederverwendbare Konstrukte AWS Cloud Development Kit (AWS CDK) und anderen modularisierten Code zu erstellen.

Verwenden von AWS Serverless Application Model (AWS SAM) als IaC-Tool

Das [AWS Serverless Application Model \(AWS SAM\)](#) ist ein Toolkit, das erweitert wird. AWS CloudFormation Es enthält zusätzliche Funktionen, mit denen Sie serverlose Anwendungen schneller erstellen können. Wenn Sie eine AWS SAM Vorlage bereitstellen, wird sie CloudFormation in eine Vorlage konvertiert, um die definierten Ressourcen zu erstellen. AWS SAM besteht aus zwei Teilen, der [AWS SAM Template-Spezifikation](#) und der [AWS SAM Command Line Interface \(AWS SAM CLI\)](#). Sie können die CloudFormation Syntax zwar direkt in der AWS SAM Vorlage verwenden, AWS SAM bietet jedoch eine eigene, einzigartige Syntax, die sich speziell auf die Beschleunigung der serverlosen Entwicklung konzentriert. Diese Kurzsyntax ermöglicht optimierte Definitionen von IaC für serverlose Ressourcen wie Amazon API Gateway und AWS Lambda Ressourcen. AWS Step Functions Die AWS SAM CLI ist ein Entwicklertool, das Funktionen enthält, mit denen Sie AWS Lambda Funktionen lokal testen, CI/CD-Pipelines (Continuous Integration and Continuous Delivery) erstellen und Befehle zur Bereitstellung serverloser Anwendungen ausführen können.

Vorteile der Verwendung von: AWS SAM

- AWS SAM hat die gleichen Vorteile wie CloudFormation.
- Im Vergleich zu CloudFormation können Sie AWS SAM damit einfacher serverlose Anwendungen und Ressourcen erstellen, z. B. ein Amazon API Gateway, das von AWS Lambda unterstützt wird.
- Mit der AWS SAM CLI können Sie AWS Lambda Funktionen lokal testen. Wenn Sie lokal eine Lambda-Funktion im Debug-Modus aufrufen, können Sie ihr anschließend einen Debugger hinzufügen. Mit dem Debugger können Sie Ihren Code Zeile für Zeile durchgehen, die Werte verschiedener Variablen einsehen und Probleme beheben, genauso wie Sie es bei jeder anderen Anwendung tun würden.

Nachteile der Verwendung AWS SAM von:

- AWS SAM hat die gleichen Nachteile wie CloudFormation.
- AWS SAM kann nicht außerhalb von verwendet werden AWS.

Verwendung des AWS CDK als IaC-Tool

Das [AWS Cloud Development Kit \(AWS CDK\)](#) ist ein Open-Source-Framework für die Softwareentwicklung, mit dem Sie Ihre Cloud-Anwendungsressourcen mithilfe vertrauter Programmiersprachen definieren können. Das AWS CDK unterstützt Python JavaScript TypeScript, Java, C# und Go. Das AWS CDK stellt Ihre Ressourcen auf sichere und wiederholbare Weise bereit. AWS CloudFormation Wenn Sie Ihren AWS CDK Code synthetisieren, ist das Ergebnis eine CloudFormation Vorlage. Das AWS CDK bietet Abstraktionen auf hoher Ebene, die den Prozess der Definition AWS von Ressourcen vereinfachen.

Das AWS CDK verwendet das Konzept der [Konstrukte](#). Ein Konstrukt ist eine Komponente innerhalb Ihrer Anwendung, die eine oder mehrere CloudFormation Ressourcen und deren Konfiguration darstellt, z. B. einen Amazon Simple Storage Service (Amazon S3) -Bucket. Konstrukte können zusammengestellt und angepasst werden, um eine komplexere Infrastruktur zu schaffen. Weitere Informationen finden Sie in der AWS CDK Dokumentation unter [Konstruktebenen](#). Das AWS CDK generiert CloudFormation Vorlagen auf der Grundlage des von Entwicklern geschriebenen Codes. Dadurch entfällt die Notwendigkeit einer manuellen CloudFormation Vorlagengenerierung. Viele Organisationen passen Konstrukte innerhalb einer Community an, teilen sie und verwenden sie wieder, genau wie jede andere Softwarebibliothek. Das Teilen von Konstrukten hilft Entwicklern dabei, schneller zu programmieren und bewährte Methoden standardmäßig zu integrieren.

AWS CDK [Aspekte](#) können Organisationen dabei helfen, Standards auf alle Konstrukte innerhalb eines bestimmten Bereichs anzuwenden. Der Aspekt könnte die Konstrukte modifizieren, beispielsweise durch Hinzufügen von Tags. Oder es könnte etwas über den Zustand der Konstrukte überprüfen.

Auf diese AWS CDK Weise können Entwickler ihre vorhandenen Programmierkenntnisse und -kenntnisse nutzen, um die Cloud-Infrastruktur zu definieren. Durch die Verwendung vertrauter Programmiersprachen können Entwickler ihr Fachwissen zur Beschreibung von AWS Ressourcen einsetzen, was den Übergang von der Anwendungsentwicklung zur Infrastrukturbereitstellung erleichtert. AWS CDK Sie können auch den Aufbau der Infrastruktur beschleunigen. AWS Dies beschleunigt den Entwicklungszyklus im Vergleich zum manuellen Schreiben CloudFormation von Vorlagen.

Vorteile der Verwendung von AWS CDK:

- Das AWS CDK unterstützt bekannte Programmiersprachen.
- Allzweckssprachen ermöglichen die Verwendung logischer Konstrukte wie For-Schleifen, Objekte, starke Typen und andere Programmiertechniken. Dies hilft Entwicklern, die Infrastruktur präzise und fehlerfrei zu deklarieren. Dieser Ansatz ermöglicht es auch, eine integrierte Entwicklungsumgebung (IDE) und zugehörige Tools zu verwenden, um die Komplexität im Zusammenhang mit der Deklaration einer großen Anzahl von Ressourcen zu bewältigen.
- AWS CDK Konstrukte können gemeinsam genutzt werden und helfen Ihnen, Ihre Governance- und Compliance-Anforderungen zu erfüllen.
- Die AWS CDK Konstrukte können den Zeit- und Arbeitsaufwand für die Entwicklung verringern. Weitere Informationen finden Sie in der [API-Referenz zur Construct Library](#).
- Das AWS CDK basiert auf CloudFormation. Wenn Sie mit den CloudFormation Konzepten vertraut sind, sind AWS CDK Konzepte leichter zu verstehen.
- Das AWS CDK hilft Ihnen bei der Durchführung von [Unit-Tests und Snapshot-Tests](#).
- Wenn eine Funktion nicht nativ in der unterstützt wird AWS CDK, können Sie ein [Level-1-Konstrukt](#) und [Rohüberschreibungen](#) verwenden. Alternativ können Sie eine [CloudFormation benutzerdefinierte Ressource](#) verwenden, die die API direkt aufruft.
- Sie können Ressourcen effizient bereinigen, indem Sie CloudFormation Stapel löschen.

Nachteile der Verwendung von: AWS CDK

- Das AWS CDK erfordert jeweils eine [Bootstrap-Umgebung](#). AWS-Konto Bootstrapping ist eine einmalige Aktion, die Sie für jede Umgebung ausführen müssen, in der Sie Ressourcen bereitstellen.
- Das AWS CDK kann zur Bereitstellung von IaC nur in der verwendet werden. AWS Cloud

Verwendung von Terraform als IaC-Tool für AWS Cloud

[HashiCorp Terraform](#) ist ein Infrastructure-as-Code-Tool (IaC), mit dem Sie Ihre Cloud-Infrastruktur verwalten können. Mit Terraform können Sie sowohl Cloud- als auch lokale Ressourcen in Konfigurationsdateien definieren, die Sie versionieren, wiederverwenden und teilen können. Anschließend können Sie einen konsistenten Workflow verwenden, um Ihre gesamte Infrastruktur während ihres gesamten Lebenszyklus bereitzustellen und zu verwalten.

Entwickler verwenden eine übergeordnete Konfigurationssprache namens [Terraform](#). Die native Low-Level-Syntax der Terraform-Sprache ist [HashiCorpConfiguration Language \(HCL\)](#). Die Terraform-Sprache ist so konzipiert, dass sie für Menschen leicht zu lesen und zu schreiben ist. Sie verwenden die Terraform-Sprache, um den gewünschten Endzustand der Cloud- oder lokalen Infrastruktur zu beschreiben. Terraform generiert dann einen Plan zum Erreichen dieses Endzustands, und Sie führen den Plan zur Bereitstellung der Infrastruktur aus.

Vorteile der Verwendung von Terraform:

- Terraform ist plattformunabhängig. Sie können es mit jedem Cloud-Dienstanbieter verwenden. Sie können die Infrastruktur bei und vielen anderen Cloud-Anbietern konfigurieren, testen AWS und bereitstellen. Wenn Ihr Unternehmen mehrere Cloud-Anbieter verwendet, kann Terraform eine einzige, einheitliche und konsistente Lösung für die Verwaltung der Cloud-Infrastruktur sein. Weitere Informationen zur Multi-Cloud-Unterstützung finden Sie unter [Multi-Cloud-Bereitstellung](#) auf der Terraform-Website.
- Terraform ist agentenlos. Es muss keine Software auf der verwalteten Infrastruktur installiert werden.
- Terraform-Module sind eine leistungsstarke Methode zur Wiederverwendung von Code und zur Einhaltung des Prinzips Don't Repeat Yourself (DRY). Beispielsweise haben Sie möglicherweise eine spezifische Konfiguration für eine Anwendung, die eine Amazon Elastic Compute Cloud (Amazon EC2) -Instance, Amazon Elastic Block Store (Amazon EBS) -Volumes und andere Ressourcen enthält, die logisch gruppiert sind. Wenn Sie mehrere Kopien dieser Konfiguration oder Anwendung erstellen müssen, können Sie die Ressourcen in ein Terraform-Modul packen und mehrere Instanzen des Moduls erstellen, anstatt den gesamten Code mehrmals zu kopieren. Diese Module können Ihnen helfen, Konfigurationen zu organisieren, zu kapseln und wiederzuverwenden. Sie sorgen auch für Konsistenz und gewährleisten bewährte Verfahren.
- Terraform ist in der Lage, [Abweichungen \(Terraform-Blogbeitrag\) in Ihrer Infrastruktur zu erkennen und zu verwalten](#). Wenn beispielsweise von Terraform verwaltete Ressourcen außerhalb von Terraform geändert werden, können Sie die Abweichung erkennen und sie mithilfe der Terraform-CLI in den gewünschten Zustand zurückversetzen.

Nachteile der Verwendung von Terraform:

- Support für neue Funktionen oder neue Ressourcen im Zusammenhang mit Cloud-Anbietern ist möglicherweise nicht verfügbar.

- Terraform verwaltet Ihren Bundesstaat nicht automatisch wie. AWS CloudFormation Es wird standardmäßig in einer lokalen Datei gespeichert, aber Sie können es auch remote in einem [Amazon S3 S3-Bucket](#) oder über [Terraform](#) Enterprise speichern.
- Der Terraform-Status kann vertrauliche Daten wie Datenbankkennwörter enthalten, die Sicherheitsbedenken aufwerfen können. Es hat sich bewährt, Ihre Statusdatei zu verschlüsseln, sie remote zu speichern, die Dateiversionierung für sie zu aktivieren und die geringsten Rechte für Lese- und Schreibvorgänge zu verwenden. Weitere Informationen finden Sie unter [Schützen vertraulicher Daten mithilfe von AWS Secrets Manager und HashiCorp Terraform](#).
- [Im August 2023 kündigte Hashicorp an, dass es nicht mehr als Open Source unter der Mozilla Public License lizenziert werden würde](#). Stattdessen ist es jetzt unter der [Business](#) Source License lizenziert.

Verwendung von Pulumi als IaC-Tool für die AWS Cloud

[Pulumi](#) ist ein Open-Source-Tool für Infrastruktur als Code. Es unterstützt bestehende, gängige Programmiersprachen wie, TypeScript, Python JavaScript, Go, .NET, Java und YAML. Es nutzt auch sein natives Ökosystem, um über das Pulumi SDK mit Cloud-Ressourcen zu interagieren. Eine herunterladbare CLI, Runtime, Bibliotheken und ein gehosteter Service arbeiten zusammen, um die Cloud-Infrastruktur bereitzustellen, zu aktualisieren und zu verwalten.

Sie können das Pulumi SDK verwenden, um Cloud-Software zu erstellen und bereitzustellen, die Container, serverlose Funktionen, gehostete Dienste und Infrastruktur in jeder Cloud verwendet.

Sie können Pulumi optional mit der Pulumi Cloud koppeln. Pulumi Cloud ist ein verwalteter Dienst, der Ihren Status und Ihre Geheimnisse speichert und die Bereitstellungen Ihrer Pulumi-Infrastruktur verwaltet.

Vorteile der Verwendung von Pulumi:

- Pulumi bietet Infrastruktur von mehr als fünfzig Cloud- und Software-as-a-Service (SaaS) - Anbietern.
- Es bietet eine vollständige und konsistente Oberfläche, die darauf ausgelegt ist, die Komplexität der Cloud zu reduzieren.

Nachteile der Verwendung von Pulumi:

- Pulumi hat zwar eine aktive Community, die Support anbietet, aber sie ist kleiner als die Terraform-Community.
- Pulumi hat eine steilere Lernkurve.

Auswahl eines IaC-Tools

Welches Tool sollten Sie also wählen?

Bei so vielen verschiedenen Tooloptionen und unterschiedlichen Geschäftsanforderungen gibt es keinen one-size-fits-all Ansatz. Zusätzlich zu den Vor- und Nachteilen der einzelnen Tools, die in diesem Handbuch behandelt werden, sollten Sie die folgenden Empfehlungen für Ihre Geschäftsanforderungen und Ihr Betriebsmodell berücksichtigen:

- Wenn Sie eine serverlose AWS Lösung mit minimalen Abhängigkeiten oder Abhängigkeiten verwalten oder bereitstellen, ist AWS Serverless Application Model (AWS SAM) möglicherweise eine gute Option für Sie. Es hat dieselben Funktionen wie AWS CloudFormation. Es vereinfacht auch das Testen und Bereitstellen von serverlosen Anwendungen auf dem AWS Cloud.
- Wenn Sie Ihre Infrastruktur vollständig auf dem System verwalten AWS, dann AWS CloudFormation und das AWS Cloud Development Kit (AWS CDK) sind gute Optionen. Sie bieten out-of-the-box Statusverwaltung, und Sie können neue Funktionen oder AWS Ressourcen auch nativ verwenden.
- Wenn Sie ein Programm mit mehreren Anbietern suchen, insbesondere für die Verwaltung von Multi-Cloud- oder Hybrid-Cloud-Infrastrukturen, ist Terraform möglicherweise eine gute Wahl, da es plattformunabhängig ist. Mit Terraform können Sie auch eine Vielzahl von Plugins verwenden und es gibt eine große Community mit Support-Optionen für Unternehmen.
- Wenn Sie über eine Top-down-Distribution mit bewährten Methoden verfügen und wenn Sie über eine Orchestrierung verfügen, bei der Sie wiederverwendbare Module mithilfe gängiger Programmiersprachen erstellen, veröffentlichen und verteilen, ist AWS CDK dies möglicherweise eine gute Option.
- Wenn Ihr Unternehmen ein hohes Risiko tolerieren kann und Multi-Cloud- oder Hybrid-Cloud-Umgebungen unterstützen muss, sollten Sie Pulumi in Betracht ziehen.

Nächste Schritte

In diesem Leitfaden wurden die Vor- und Nachteile verschiedener Infrastructure-as-Code-Tools (IaC) erörtert, mit denen Sie AWS Ressourcen und Infrastruktur erstellen, bereitstellen und verwalten können. Je nachdem, für welches Tool Sie sich entscheiden, empfehlen wir Ihnen, zunächst die folgenden Ressourcen aufzusuchen.

AWS Cloud Development Kit (AWS CDK)

- [AWS CDK Einführungsworkshop](#)

AWS CloudFormation

- [AWS CloudFormation Labor](#)
- [AWS CloudFormation -Workshop](#)

HashiCorp Terraform

- [AWS Terraform-Werkstatt](#)
- [CDK für das Terraform-Repository](#) () GitHub

Pulumi

- [Pulumi-Endlager](#) () GitHub
- [AWS Modernisierung mit Pulumi-Werkstatt](#)

Ressourcen

AWS Dokumentation

- [Bewährte Verfahren für die Verwendung von AWS CDK in TypeScript zur Erstellung von IaC-Projekten](#) (AWS Prescriptive Guidance)
- [Suchen Sie in AWS CDK Anwendungen oder CloudFormation Vorlagen nach bewährten Verfahren, indem Sie cdk-nag-Regelpakete \(Prescriptive Guidance\) verwenden](#)AWS
- [Einführung in DevOps on AWS: Infrastruktur](#) als Code (Whitepaper)AWS
- [AWS CloudFormation -Dokumentation](#)

Weitere Dokumentation

- [Erste Schritte mit Infrastructure as Code](#) (HashiCorpWebsite)
- [HashiCorpTerraform-Dokumentation](#)
- [Pulumi-Dokumentation](#)
- [Pulumi: Was ist Infrastruktur als Code](#) (Pulumi-Website)

Tools

- [cdk-nag](#) () GitHub
- [cfn-nag](#) () GitHub
- [Terratest](#)

Mitwirkende

Inhaltserstellung

- Siamak Heshmati, leitender Architekt, DevOps AWS
- Mason Cahill, leitender Berater, DevOps AWS
- Sandip Gangapadhyay, leitender Architekt, DevOps AWS
- Sandeep Gawande, leitender Berater, AWS
- Rajneesh Tyagi, Leitender Berater, AWS

Überprüfung

- David Howerton, leitender Engagementmanager, AWS
- Steven Guggenheimer, leitender Architekt für Cloud-Anwendungen, AWS

Technisches Schreiben

- Lilly AbouHarb, leitende technische Redakteurin, AWS

Dokumentverlauf

In der folgenden Tabelle werden wichtige Änderungen in diesem Leitfaden beschrieben. Um Benachrichtigungen über zukünftige Aktualisierungen zu erhalten, können Sie einen [RSS-Feed](#) abonnieren.

Änderung	Beschreibung	Datum
Pulumi-Updates	Wir haben das Pulumi-Kapitel aktualisiert.	17. Februar 2026
Erste Veröffentlichung	—	23. Februar 2024

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.