AWS-Whitepaper

Einführung in DevOps in AWS

Einführung in DevOps in AWS: AWS-Whitepaper

Table of Contents

Einführung in DevOps in AWS

Veröffentlichungsdatum: 16. Oktober 2020 (Dokumentversionen)

Überblick

Unternehmen begeben sich heute mehr denn je auf den Weg der digitalen Transformation, um engere Beziehungen mit ihren Kunden zu entwickeln und so einen nachhaltigen und dauerhaften Geschäftswert zu erzielen. Unternehmen aller Formen und Größen setzen ihre Mitbewerber massiv unter Druck und erschließen durch schneller denn je umgesetzte Innovationen neue Märkte. Für diese Organisationen ist es wichtig, sich auf Innovation und Softwareunterbrechung zu konzentrieren. Daher ist es unerlässlich, ihre Softwarebereitstellung zu rationalisieren. Unternehmen, die ihre Zeit von der Idee bis zur Produktion verkürzen und Geschwindigkeit und Agilität vorrangig behandeln, könnten die radikalen Erneuerer von morgen sein.

Es gibt mehrere Faktoren, die berücksichtigt werden müssen, um der nächste digitale radikale Erneuerer zu werden. Dieses Whitepaper konzentriert sich jedoch auf DevOps und die Services und Funktionen der AWS-Plattform, die dazu beitragen, die Fähigkeit eines Unternehmens zu verbessern, Anwendungen und Services mit hoher Geschwindigkeit bereitzustellen.

Einführung

DevOps ist die Kombination aus kulturellen, technischen Verfahren und Mustern sowie Tools, die es einer Organisation ermöglichen, Anwendungen und Services mit hoher Geschwindigkeit und besserer Qualität bereitzustellen. Im Laufe der Zeit haben sich bei der Einführung von DevOps mehrere wichtige Verfahren herausgebildet: Continuous Integration (kontinuierliche Integration), Continuous Delivery (kontinuierliche Bereitstellung), Infrastructure as Code sowie Überwachung und Protokollierung.

In diesem Dokument werden die AWS-Funktionen vorgestellt, mit denen Sie Ihre DevOps-Reise beschleunigen können. Außerdem wird beschrieben, wie AWS-Services dazu beitragen können, den undifferenzierten Arbeitsaufwand im Zusammenhang mit der DevOps-Anpassung zu beseitigen. Wir zeigen auch, wie Sie die Fähigkeit zu Continuous Integration und Continuous Delivery entwickeln können, ohne Server zu verwalten oder Knoten zu erstellen, und wie Sie Infrastructure as Code nutzen können, um Ihre Cloud-Ressourcen konsistent und wiederholbar bereitzustellen und zu verwalten.

- Continuous Integration (kontinuierliche Integration) ist eine Praxis in der Softwareentwicklung, bei der Entwickler alle Codeänderungen regelmäßig in einem zentralen Repository zusammenführen. Anschließend werden automatisierte Builds und Tests ausgeführt.
- Continuous Delivery (kontinuierliche Bereitstellung) ist eine Softwareentwicklungsmethode, bei der Codeänderungen automatisch erstellt, getestet und für eine Produktionsversion vorbereitet werden.
- Infrastructure as Code ist eine Methode, bei der Infrastruktur unter Verwendung von Code und Softwareentwicklungsmethoden wie Versionskontrolle und Continuous Integration bereitgestellt und verwaltet wird.
- Überwachung und Protokollierung ermöglichen es Organisationen, zu ermitteln, wie Anwendungs- und Infrastrukturleistung die Endbenutzererfahrung beeinflussen.
- Kommunikation und Zusammenarbeit werden anhand von Methoden eingeführt, die die Teams näher zusammenbringen, Workflows entwickeln und die Verantwortlichkeiten für DevOps verteilen.
- Sicherheit sollte ein Querschnittsanliegen sein. Ihre Pipelines für Continuous Integration und Continuous Delivery (CI/CD) und zugehörige Services sollten geschützt werden. Außerdem sollten angemessene Zugriffssteuerungsberechtigungen eingerichtet werden.

Eine Untersuchung jedes dieser Prinzipien zeigt einen engen Zusammenhang mit den Angeboten von Amazon Web Services (AWS).

Fortlaufende Integration

Continuous Integration (CI) ist eine Softwareentwicklungsmethode, bei der Entwickler ihre Codeänderungen regelmäßig in einem zentralen Repository zusammenführen. Anschließend werden automatisierte Builds und Tests ausgeführt. CI hilft dabei, Bugs schneller zu entdecken und zu beheben, die Softwarequalität zu optimieren und den Zeitraum zu minimieren, in dem neue Softwareaktualisierungen validiert und eingeführt werden.

AWS bietet die folgenden Services für Continuous Integration:

Themen

- AWS CodeCommit
- AWS CodeBuild
- AWS CodeArtifact

AWS CodeCommit

AWS CodeCommit ist ein sicherer, hochgradig skalierbarer und verwalteter Service für Quellcodeüberwachung, der private Git-Repositorys hostet. Mit CodeCommit müssen Sie kein eigenes Quellcodeverwaltungssystem betreiben. Außerdem muss keine Hardware bereitgestellt und skaliert oder Software installiert, konfiguriert und betrieben werden. Sie können mit CodeCommit alles speichern, von Code bis zu Binärdateien. Es unterstützt die Standardfunktionalität von Git, Sie können also nahtlos mit Ihren vorhandenen Git-basierten Tools arbeiten. Ihr Team kann zum Durchsuchen, Bearbeiten und Zusammenarbeiten bei Projekten auch die Online-Code-Tools von CodeCommit verwenden. AWS CodeCommit bietet mehrere Vorteile:

Zusammenarbeit: AWS CodeCommit wurde für die kollaborative Softwareentwicklung entworfen. Sie können Ihren Code problemlos einspielen, verzweigen und zusammenführen. So behalten Sie ganz einfach die Kontrolle über die Projekte Ihres Teams. CodeCommit unterstützt auch Pull-Anfragen, die über einen Mechanismus verfügen, über den Codeüberprüfungen angefordert werden können und Code mit Kollegen diskutiert werden kann.

Verschlüsselung: Sie können Ihre Dateien wie gewünscht mithilfe von HTTPS oder SSH nach und aus AWS CodeCommit übertragen. Ihre Repositorys werden über AWS Key Management Service (AWS KMS) unter Verwendung kundenspezifischer Schlüssel im Ruhezustand automatisch verschlüsselt.

Zugriffskontrolle: AWS CodeCommit verwendet AWS Identity and Access Management (IAM), um zu kontrollieren und zu überwachen, wer auf Ihre Daten zugreifen kann und wie, wann und wo der Zugriff erfolgen kann. CodeCommit hilft Ihnen auch dabei, Ihre Repositorys über AWS CloudTrail und Amazon CloudWatch zu überwachen.

Hohe Verfügbarkeit und Beständigkeit: AWS CodeCommit speichert Ihre Repositorys in Amazon Simple Storage Service (Amazon S3) und Amazon DynamoDB. Ihre verschlüsselten Daten werden redundant an mehreren Standorten gespeichert. Durch diese Architektur wird die Verfügbarkeit und Beständigkeit Ihrer Repository-Daten erhöht.

Benachrichtigungen und benutzerdefinierte Skripte: Sie können Benachrichtigungen für Ereignisse erhalten, die sich auf Ihre Repositorys auswirken. Benachrichtigungen werden als Amazon Simple Notification Service (Amazon SNS)-Nachrichten gesendet. Jede Benachrichtigung enthält eine Statusmeldung sowie einen Link zu den Ressourcen, deren Ereignis diese Benachrichtigung ausgelöst hat. Mithilfe der Repository-Auslöser in AWS CodeCommit können Sie zudem Benachrichtigungen senden und HTTP-Webhooks mit Amazon SNS erstellen oder AWS Lambda-Funktionen zu den von Ihnen ausgewählten Repository-Ereignissen aufrufen.

AWS CodeBuild

AWS CodeBuild ist ein vollständig verwalteter Service für Continuous Integration, der Quellcode kompiliert, Tests ausführt und Softwarepakete generiert, die direkt bereitgestellt werden können. Sie brauchen keine eigenen Buildserver bereitstellen, verwalten und skalieren. CodeBuild kann entweder GitHub, GitHub Enterprise, BitBucket, AWS CodeCommit oder Amazon S3 als Quellanbieter verwenden.

CodeBuild skaliert kontinuierlich und kann mehrere Builds gleichzeitig verarbeiten. CodeBuild bietet verschiedene vorkonfigurierte Umgebungen für unterschiedliche Versionen von Microsoft Windows und Linux. Kunden können als benutzerdefinierte Buildumgebungen auch Docker-Container verwenden. CodeBuild lässt sich auch in Open-Source-Tools wie Jenkins und Spinnaker integrieren.

Außerdem kann CodeBuild Berichte für Einheiten-, Funktions- oder Integrationstests erstellen. Diese Berichte bieten einen visuellen Überblick darüber, wie viele Testfälle ausgeführt wurden und welche positiv bzw. negativ ausfielen. Der Buildprozess kann auch in einer Amazon Virtual Private Cloud (Amazon VPC) ausgeführt werden, was nützlich sein kann, wenn Ihre Integrationsservices oder Datenbanken in einer VPC bereitgestellt werden.

AWS CodeArtifact

AWS CodeArtifact ist ein vollständig verwalteter Artefakt-Repository-Service, mit dem Organisationen die in ihrem Software-Entwicklungsprozess verwendeten Softwarepakete sicher speichern, veröffentlichen und freigeben können. CodeArtifact kann für das automatische Abrufen von Softwarepaketen und Abhängigkeiten aus öffentlichen Artefakt-Repositorys konfiguriert werden, damit Entwickler stets Zugriff auf die neuesten Versionen haben.

Softwareentwicklerteams verlassen sich zunehmend auf Open-Source-Pakete, um allgemeine Aufgaben in ihrem Anwendungspaket auszuführen. Für die Softwareentwicklerteams ist es von entscheidender Bedeutung, die Kontrolle darüber zu behalten, dass eine bestimmte Version der Open-Source-Software frei von Schwachstellen ist. Mit CodeArtifact können Sie Kontrollen einrichten, um das Vorgenannte zu erzwingen.

CodeArtifact funktioniert mit gängigen Paketmanagern und erstellt Tools wie Maven, Gradle, npm, yarn, twine und pip, wodurch die Integration in bestehende Entwicklungsworkflows einfach ist.

Fortlaufende Lieferung

Continuous Delivery (kontinuierliche Bereitstellung) ist eine Softwareentwicklungsmethode, bei der Codeänderungen automatisch erstellt, getestet und für eine Produktionsversion vorbereitet werden. Als wichtiges Element der modernen Anwendungsentwicklung ist Continuous Delivery eine Erweiterung der Continuous Integration (kontinuierliche Integration), indem nach dem Aufbaustadium alle Codeänderungen in einer Test- und/oder Produktionsumgebung bereitgestellt werden. Bei einer korrekten Implementierung steht den Entwicklern stets ein Erstellungsartefakt für die Bereitstellung zur Verfügung, das bereits einen standardisierten Testprozess durchlaufen hat.

Continuous Delivery ermöglicht Entwicklern automatische Tests, die über einfache Einheitstests hinausgehen. Dadurch können Anwendungsaktualisierungen über mehrere Dimensionen hinweg getestet werden, bevor sie für einen Kunden bereitgestellt werden. Diese Tests können UI-, Last-, Integrations- sowie API-Zuverlässigkeitstests uvm. umfassen. Sie unterstützen Entwickler bei der gründlichen Validierung von Aktualisierungen, auch Probleme können schon vorab entdeckt werden. Dank der Cloud ist die automatische Erstellung und Replikation mehrerer Testumgebungen heutzutage noch einfacher und kostengünstiger als die Durchführung dieser Prozesse an einem lokalen Standort.

AWS bietet die folgenden Services für Continuous Delivery:

- AWS CodeBuild
- AWS CodeDeploy
- AWS CodePipeline

Themen

- AWS CodeDeploy
- AWS CodePipeline

AWS CodeDeploy

AWS CodeDeploy ist ein vollständig verwalteter Bereitstellungsservice, der Softwarebereitstellungen für eine Vielzahl von Computing-Services wie Amazon Elastic Compute Cloud (Amazon EC2), AWS Fargate, AWS Lambda und Ihre On-Premises-Server automatisiert. AWS CodeDeploy erleichtert Ihnen die schnelle Veröffentlichung neuer Funktionen, hilft Ihnen, Ausfallzeiten während der

Anwendungsbereitstellung zu vermeiden, und bewältigt die Komplexität der Aktualisierung Ihrer Anwendungen. Sie können CodeDeploy für die automatisierten Softwarebereitstellungen nutzen und so fehleranfällige manuelle Vorgänge beseitigen. Der Service skaliert sich auf Basis Ihrer Bereitstellungsanforderungen.

CodeDeploy bietet mehrere Vorteile, die sich am DevOps-Prinzip Continuous Delivery (kontinuierliche Bereitstellung) orientieren:

Automatisierte Bereitstellungen: CodeDeploy bietet eine vollständige Automatisierung und dadurch Zuverlässigkeit und Schnelligkeit bei der Bereitstellung von Software.

Zentrale Kontrolle: Mit CodeDeploy können Sie problemlos über die AWS-Managementkonsole oder die AWS CLI Anwendungsbereitstellungen starten und deren Status nachverfolgen. Mit CodeDeploy erhalten Sie detaillierte Berichte, in denen Sie sehen können, wann und wo eine Anwendungsänderung bereitgestellt wurde. Sie können auch Push-Benachrichtigungen erstellen, um Live-Updates zu Ihren Bereitstellungen zu erhalten.

Ausfallzeiten minimieren: CodeDeploy unterstützt Sie bei der Optimierung der Verfügbarkeit Ihrer Anwendung während des Softwarebereitstellungsprozesses. Änderungen werden nach und nach implementiert und der Status der Anwendung wird gemäß konfigurierbarer Regeln überwacht. Falls Fehler auftreten, können Softwarebereitstellungen einfach angehalten und rückgängig gemacht werden.

Einfache Einführung: CodeDeploy funktioniert mit jeder Anwendung und bietet dieselbe Erfahrung auf verschiedenen Plattformen und in unterschiedlichen Sprachen. Sie können Ihren vorhandenen Einrichtungscode problemlos wiederverwenden. CodeDeploy lässt sich auch in Ihren vorhandenen Software-Freigabeprozess oder die Continuous Delivery Toolchain integrieren (z. B. AWS CodePipeline, GitHub, Jenkins).

AWS CodeDeploy unterstützt mehrere Bereitstellungsoptionen. Weitere Informationen finden Sie unter Bereitstellungsstrategien.

AWS CodePipeline

AWS CodePipeline ist ein Continuous-Delivery-Service, mit dem Sie die für die Freigabe Ihres Codes erforderlichen Schritte entwickeln, visualisieren und automatisieren können. Mit AWS CodePipeline können Sie den vollständigen Veröffentlichungsprozess für die Entwicklung Ihres Codes modellieren, Vorproduktionsumgebungen bereitstellen, Ihre Anwendung testen und sie zur Produktion freigeben. Anschließend erstellt und testet AWS CodePipeline Ihre Anwendung und stellt sie bereit. Dieser

Prozess läuft bei jeder Codeänderung nach dem definierten Workflow ab. Sie können in jeder Phase des Release-Prozesses APN-Partnertools oder Ihre eigenen benutzerdefinierten Tools integrieren, um eine umfassende Continuous-Delivery-Lösung zu erstellen.

AWS CodePipeline bietet mehrere Vorteile, die sich am DevOps-Prinzip Continuous Delivery (kontinuierliche Bereitstellung) orientieren:

Schnelle Bereitstellung: AWS CodePipeline automatisiert Ihren Software-Release-Prozess und ermöglicht Ihnen die schnelle Veröffentlichung neuer Funktionen für Benutzer. Mit CodePipeline können Sie schnell auf Feedback reagieren und die neuen Funktionen den Benutzern schneller bereitstellen.

Verbesserte Qualität: Wenn Sie die Prozesse zur Entwicklung, zum Testen und zur Veröffentlichung automatisieren, können Sie dank AWS CodePipeline die Geschwindigkeit und Qualität Ihrer Software-Updates erhöhen, da alle neuen Änderungen die gleichen Qualitätsprüfungen durchlaufen.

Einfache Integration: AWS CodePipeline kann einfach erweitert und so an Ihre spezifischen Bedürfnisse angepasst werden. Sie können in jeder Phase des Release-Prozesses unsere vorgefertigten Plug-Ins oder eigene benutzerdefinierte Plug-Ins verwenden. Sie können beispielsweise Ihren Quellcode aus GitHub abrufen, Ihren lokalen Jenkins-Build-Server verwenden, Belastungstests über einen Drittanbieterservice ausführen oder Bereitstellungsinformationen an Ihr benutzerdefiniertes Prozess-Dashboard weiterleiten.

Konfigurierbarer Workflow: Mit AWS CodePipeline können Sie die verschiedenen Phasen Ihres Software-Release-Prozesses mithilfe der Konsolenoberfläche, der AWS CLI, AWS CloudFormation oder den AWS SDKs modellieren. Sie können die ausgeführten Tests und die Schritte zur Bereitstellung und individuellen Anpassung Ihrer Anwendung und der zugehörigen Abhängigkeiten einfach angeben.

Bereitstellungsstrategien

Bereitstellungsstrategien definieren, wie Sie Ihre Software bereitstellen möchten. Organisationen verfolgen basierend auf ihrem Geschäftsmodell unterschiedliche Bereitstellungsstrategien. Einige entscheiden sich möglicherweise dafür, Software bereitzustellen, die vollständig getestet wurde. Andere möchten möglicherweise, dass ihre Benutzer Feedback geben und Funktionen in der Entwicklungsumgebung (z. B. Beta-Versionen) bewerten. Im folgenden Abschnitt stellen wir verschiedene Bereitstellungsstrategien vor.

Themen

In-Situ-Bereitstellungen

Bei dieser Bereitstellungsstrategie wird die Anwendung in jeder Instance in der Bereitstellungsgruppe beendet, die neueste Anwendungsrevision installiert und die neue Version der Anwendung gestartet und validiert. Sie können einen Load Balancer verwenden, damit jede Instance während ihrer Bereitstellung abgemeldet und nach Abschluss der Bereitstellung wiederhergestellt wird. In-Situ-Bereitstellungen können auf einmal erfolgen, sofern ein Serviceausfall vorliegt, oder als paralleles Update durchgeführt werden. AWS CodeDeploy und AWS Elastic Beanstalk bieten Bereitstellungskonfigurationen für jeweils eine Instance, für die Hälfte der Instances und für alle Instances auf einmal. Diese Strategien für In-Situ-Bereitstellungen sind in Blau/Grün-Bereitstellungen verfügbar.

Blau/Grün-Bereitstellungen

Eine Blau/Grün-Bereitstellung, manchmal auch als Rot/Schwarz bezeichnet, ist eine Methode zur Freigabe von Anwendungen, wobei der Datenverkehr zwischen zwei identischen Umgebungen mit verschiedenen Anwendungsversionen verlagert wird. Blau/Grün-Bereitstellungen helfen Ihnen, Ausfallzeiten während Anwendungsupdates und Risiken im Zusammenhang mit Ausfallzeiten und Rollback-Funktionen zu minimieren. Blau/Grün-Bereitstellungen ermöglichen es Ihnen, eine

neue Version (grün) Ihrer Anwendung zusammen mit der alten Version (blau) zu starten und die neue Version zu überwachen und zu testen, bevor Sie den Datenverkehr dorthin umleiten und bei Erkennung eines Problems ein Rollback durchführen müssen.

Canary-Bereitstellungen

Der Datenverkehr wird in zwei Inkrementen verschoben. Eine Canary-Bereitstellung ist eine Blau/Grün-Strategie, die risikoscheuer ist und bei der ein stufenweiser Ansatz verwendet wird. Die Strategie kann zweistufig oder linear sein, wobei neuer Anwendungscode bereitgestellt und zum Testen zur Verfügung gestellt wird. Nach Annahme wird der Code entweder für den Rest der Umgebung oder linear eingeführt.

Lineare Bereitstellungen

Bei linearen Bereitstellungen wird der Verkehr in gleichen Inkrementschritten mit einer gleichen Anzahl von Minuten zwischen den einzelnen Inkrementschritten verschoben. Sie können aus vordefinierten linearen Optionen wählen, die den prozentualen Anteil des Datenverkehrs angeben, der in jedem Inkrementschritt verschoben wird, sowie die Anzahl der Minuten zwischen den einzelnen Inkrementschritten.

Gleichzeitige Bereitstellungen von allem

Gleichzeitige Bereitstellung von allem bedeutet, dass der gesamte Datenverkehr auf einmal von der ursprünglichen in die neue Umgebung verlagert wird.

Matrix für Bereitstellungsstrategien

In der folgenden Matrix sind die unterstützten Bereitstellungsstrategien für Amazon Elastic Container Service (Amazon ECS) AWS Lambda und Amazon EC2/On-Premise aufgeführt.

- Amazon ECS ist ein vollständig verwalteter Orchestrierungsservice.
- Mit AWS Lambda können Sie Code ausführen, ohne dass Sie Server bereitstellen und verwalten müssen.
- Amazon EC2 ermöglicht eine sichere und anpassbare Rechenkapazität in der Cloud.

	A	B	C	D
1	Matrix für Bereitstellungsstrategien	Amazon ECS	AWS Lambda	Amazon EC2/ On-Premise
2	In-Situ	✓	✓	✓
3	Blau/Grün	✓	✓	✓*
4	Canary	✓	✓	X
5	Linear	✓	✓	X
6	Alles gleichzeitig	✓	✓	X

> **Note**
>
> Die blaue/grüne Bereitstellung mit EC2/On-Premise funktioniert nur mit EC2-Instances.

Bereitstellungsstrategien für AWS Elastic Beanstalk

AWS Elastic Beanstalk unterstützt die folgenden Bereitstellungsstrategien:

- Alles gleichzeitig: Führt eine In-Situ-Bereitstellung auf allen Instances durch.

- Fortlaufend: Teilt die Instances in Batches auf und nimmt die Bereitstellung jeweils für einen Batch vor.
- Fortlaufend mit zusätzlichem Batch: Teilt die Bereitstellungen in Batches auf, erstellt aber für den ersten Batch neue EC2-Instances, anstatt die Bereitstellung auf den vorhandenen EC2-Instances vorzunehmen.
- Unveränderlich: Verwenden Sie diese Strategie, wenn Sie eine Bereitstellung mit einer neuen Instance vornehmen müssen, anstatt eine vorhandene Instance zu verwenden.
- Datenverkehrsaufteilung: Führt eine unveränderliche Bereitstellung durch und leitet dann einen Prozentsatz des Datenverkehrs für eine vorher festgelegte Dauer an die neuen Instances weiter. Wenn die Instances fehlerfrei bleiben, wird der gesamte Datenverkehr an neue Instances weitergeleitet und alte Instances werden heruntergefahren.

Infrastruktur als Code

Ein grundsätzliches Prinzip von DevOps besteht darin, Infrastruktur so zu behandeln, wie Entwickler Code behandeln. Anwendungscode hat ein definiertes Format und eine definierte Syntax. Wenn der Code nicht gemäß den Regeln der Programmiersprache geschrieben wurde, können keine Anwendungen erstellt werden. Code wird in einem Versionsverwaltungs- oder Quellcodeverwaltungssystem gespeichert, das den Codeentwicklungsverlauf, Änderungen und Fehlerkorrekturen protokolliert. Wenn Code kompiliert oder in Anwendungen integriert wird, erwarten wir, dass eine konsistente Anwendung erstellt wird und der Build wiederholbar und zuverlässig ist.

Der Einsatz von Infrastructure as Code bedeutet, die gleiche Präzision der Anwendungscodeentwicklung auch auf die Infrastrukturbereitstellung anzuwenden. Alle Konfigurationen sollten deklarativ definiert und wie Anwendungscode in einem Quellcodeverwaltungssystem wie AWS CodeCommit gespeichert werden. Die Orchestrierung und Bereitstellung der Infrastruktur sollte auch die Verwendung von Infrastructure as Code unterstützen.

Die Infrastruktur wurde traditionell mit einer Kombination aus Skripten und manuellen Prozessen bereitgestellt. Manchmal wurden diese Skripte in Versionskontrollsystemen gespeichert oder Schritt für Schritt in Textdateien oder Runbooks dokumentiert. Oft ist die Person, die die Runbooks schreibt, nicht dieselbe Person, die diese Skripte ausführt oder die Runbooks durchgeht. Wenn diese Skripte oder Runbooks nicht regelmäßig aktualisiert werden, können sie in Bereitstellungen zu einem Show-Stopper werden. Dies führt dazu, dass die Schaffung neuer Umgebungen nicht immer wiederholbar, zuverlässig oder konsistent ist.

Im Gegensatz dazu bietet AWS eine DevOps-fokussierte Methode zur Erstellung und Wartung der Infrastruktur. Ähnlich wie Software-Entwickler Anwendungscode schreiben, stellt AWS Services bereit, die die Erstellung, Bereitstellung und Wartung der Infrastruktur in einer programmgesteuerten, beschreibenden und erläuternden Weise ermöglicht. Diese Services bieten Präzision, Klarheit und Zuverlässigkeit. Die in diesem Dokument erörterten AWS-Services sind der Kern einer DevOps-Methodik und bilden die Grundlage zahlreicher übergeordneter AWS-DevOps-Prinzipien und -Praktiken.

AWS bietet folgende Services an, um Infrastructure as Code zu definieren.

- AWS CloudFormation
- AWS Cloud Development Kit (AWS CDK)
- AWS Cloud Development Kit for Kubernetes

AWS CloudFormation

AWS CloudFormation ist ein Service, mit dem Entwickler AWS-Ressourcen geordnet und vorhersehbar erstellen können. Ressourcen werden in Textdateien im JavaScript Object Notation (JSON)- oder Yet Another Markup Language (YAML)-Format geschrieben. Die Vorlagen erfordern abhängig von der Art der Ressourcen, die erstellt und verwaltet werden, eine bestimmte Syntax und Struktur. Sie erstellen Ihre Ressourcen in JSON oder YAML mit einem beliebigen Code-Editor wie AWS Cloud9 und checken sie in ein Versionskontrollsystem ein. CloudFormation erstellt dann die angegebenen Services auf sichere, wiederholbare Weise.

Eine CloudFormation-Vorlage wird in der AWS-Umgebung als Stack bereitgestellt. Sie können Stacks über die AWS-Managementkonsole, AWS Command Line Interface oder AWS-CloudFormation-APIs verwalten. Wenn Sie Änderungen an den ausgeführten Ressourcen in einem Stack vornehmen müssen, aktualisieren Sie den Stack. Bevor Sie Änderungen an Ihren Ressourcen vornehmen, können Sie einen Änderungssatz generieren, der eine Zusammenfassung der von Ihnen vorgeschlagenen Änderungen darstellt. Anhand von Änderungssätzen können Sie sehen, wie sich Ihre Änderungen möglicherweise auf die ausgeführten Ressourcen auswirken, bevor Sie sie implementieren, insbesondere bei kritischen Ressourcen.

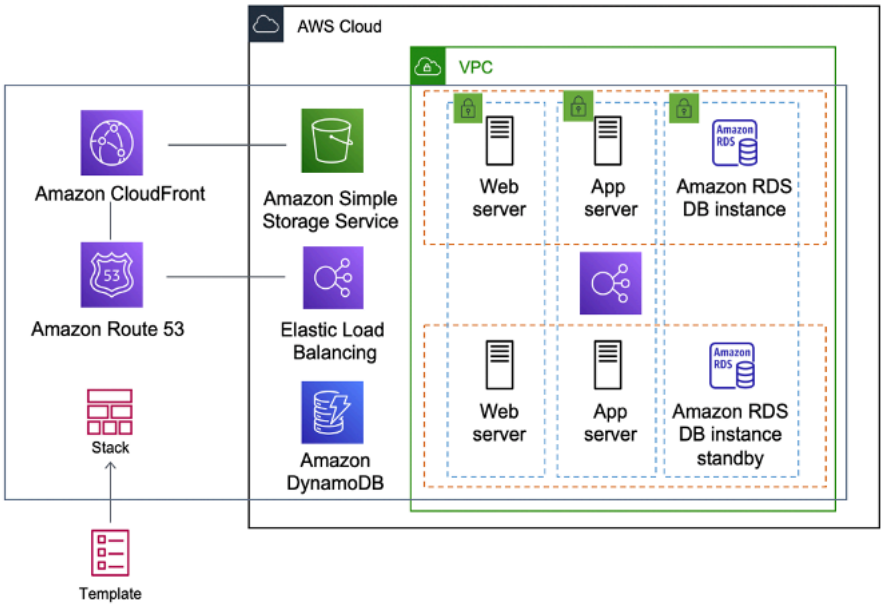

Abbildung 1: AWS CloudFormation erstellt eine gesamte Umgebung (Stack) aus einem Vorlagen-Workflow

Sie können eine einzelne Vorlage verwenden, um eine gesamte Umgebung zu erstellen und zu aktualisieren, oder separate Vorlagen, um mehrere Ebenen innerhalb einer Umgebung zu verwalten.

Auf diese Weise können Vorlagen modularisiert werden und es wird eine Governance-Ebene bereitgestellt, die für viele Organisationen wichtig ist.

Wenn Sie einen Stack in der Konsole erstellen oder aktualisieren, werden Ereignisse angezeigt, die den Status der Konfiguration anzeigen. Tritt ein Fehler auf, wird der Stack standardmäßig auf seinen vorherigen Status zurückgesetzt. Amazon Simple Notification Service (Amazon SNS) stellt Benachrichtigungen zu Ereignissen bereit. Sie können Amazon SNS beispielsweise verwenden, um die Stack-Erstellung und den Fortschritt des Löschvorgangs per E-Mail zu verfolgen und programmgesteuert in andere Prozesse zu integrieren.

AWS CloudFormation erleichtert Ihnen die Organisation und Bereitstellung einer Zusammenstellung von AWS-Ressourcen und ermöglicht Ihnen, Abhängigkeiten zu beschreiben oder besondere Parameter zu übergeben, wenn der Stack konfiguriert wird.

Mit CloudFormation-Vorlagen können Sie mit einer Vielzahl von AWS-Services wie Amazon S3, Auto Scaling, Amazon CloudFront, Amazon DynamoDB, Amazon EC2, Amazon ElastiCache, AWS Elastic Beanstalk, Elastic Load Balancing, IAM, AWS OpsWorks und Amazon VPC arbeiten. Die aktuelle Liste der unterstützten Ressourcen finden Sie unter AWS-Ressourcen- und Eigenschaftstypen – Referenz.

AWS Cloud Development Kit

AWS Cloud Development Kit (AWS CDK) ist ein Open-Source-Entwicklungsframework, das für die Modellierung und Bereitstellung Ihrer Cloud-Anwendungsressourcen auf bekannte Programmiersprachen zurückgreift. Im Rahmen von AWS CDK können Sie Anwendungsinfrastrukturen mithilfe von TypeScript, Python, Java und .NET modellieren. Entwickler können ihre vorhandene integrierte Entwicklungsumgebung (Integrated Development Environment, IDE) nutzen und Tools wie automatische Vervollständigung und Inline-Dokumentation nutzen, um die Entwicklung der Infrastruktur zu beschleunigen.

AWS CDK nutzt AWS CloudFormation im Hintergrund, um Ressourcen auf sichere, wiederholbare Weise bereitzustellen. Die Grundbausteine des CDK-Codes sind Konstrukte. Ein Konstrukt stellt eine Cloud-Komponente dar und schließt alles ein, was AWS CloudFormation zum Erstellen der Komponente benötigt. AWS CDK umfasst die AWS Construct Library, die Konstrukte enthält, die viele AWS-Services darstellen. Durch die Kombination von verschiedenen Konstrukten können Sie schnell und einfach komplexe Architekturen für die Bereitstellung in AWS erstellen.

AWS Cloud Development Kit for Kubernetes

AWS Cloud Development Kit for Kubernetes (cdk8s) ist ein Open-Source-Softwareentwicklungsframework zum Definieren von Kubernetes-Anwendungen mit universellen Programmiersprachen.

Sobald Sie Ihre Anwendung in einer Programmiersprache definiert haben (zum Zeitpunkt der Veröffentlichung werden nur Python und TypeScript unterstützt), konvertiert cdk8s Ihre Anwendungsbeschreibung in YML vor Kubernetes. Diese YML-Datei kann dann von jedem Kubernetes-Cluster verwendet werden, der irgendwo ausgeführt wird. Da die Struktur in einer Programmiersprache definiert ist, können Sie die umfangreichen Funktionen der Programmiersprache nutzen. Sie können die Abstraktionsfunktion der Programmiersprache verwenden, um Ihre eigenen Codebausteine zu erstellen und in allen Bereitstellungen wiederzuverwenden.

Automatisierung

Eine weitere Kernphilosophie und Praxis von DevOps ist die Automatisierung. Die Automatisierung konzentriert sich auf die Einrichtung, Konfiguration, Bereitstellung und Unterstützung der Infrastruktur und der darauf ausgeführten Anwendungen. Durch die Automatisierung können Sie Umgebungen schneller standardisiert und wiederholbar einrichten. Der Wegfall manueller Prozesse ist für eine erfolgreiche DevOps-Strategie entscheidend. In der Vergangenheit waren Serverkonfiguration und Anwendungsbereitstellung überwiegend manuelle Prozesse. Umgebungen sind zunehmend nicht mehr standardisiert. Eine Umgebung lässt sich demzufolge nur schwer reproduzieren, wenn Probleme auftreten.

Der Einsatz von Automatisierung ist entscheidend, um die Vorteile der Cloud voll auszuschöpfen. Intern ist AWS stark auf Automatisierung angewiesen, um die Hauptmerkmale Elastizität und Skalierbarkeit bereitzustellen. Manuelle Prozesse sind fehleranfällig, unzuverlässig und zur Unterstützung eines agilen Unternehmens unzureichend. Häufig bindet ein Unternehmen hochqualifizierte Ressourcen, um eine manuelle Konfiguration bereitzustellen. Doch die Zeit könnte besser für die Unterstützung anderer, kritischerer und wertvollerer Aktivitäten innerhalb des Unternehmens aufgewendet werden.

Moderne Betriebsumgebungen sind üblicherweise auf eine vollständige Automatisierung angewiesen, damit manuelle Eingriffe oder Zugriffe auf Produktionsumgebungen wegfallen. Dies umfasst alle Software-Releases sowie Maschinenkonfiguration, Betriebssystem-Patching, Fehlerbehebung oder Korrektur von Programmfehlern. Viele verschiedene Automatisierungspraktiken können zusammen verwendet werden, um einen automatisierten End-to-End-Prozess auf höherer Ebene bereitzustellen.

Die Automatisierung hat die folgenden Hauptvorteile:

- Schnelle Änderungen
- Verbesserte Produktivität
- Wiederholbare Konfigurationen
- Reproduzierbare Umgebungen
- Nutzung der Elastizität
- Nutzung der automatischen Skalierung
- Automatisiertes Testen

Die Automatisierung ist ein Eckpfeiler der AWS-Services und wird intern in allen Services, Funktionen und Angeboten unterstützt.

Themen

- AWS OpsWorks
- AWS Elastic Beanstalk

AWS OpsWorks

AWS OpsWorks bringt die Prinzipien von DevOps noch weiter voran als AWS Elastic Beanstalk. Der Service kann als Anwendungsverwaltungsdienst und nicht nur als Anwendungscontainer betrachtet werden. AWS OpsWorks bietet noch mehr Automatisierungsgrade mit zusätzlichen Funktionen wie die Integration in die Konfigurationsverwaltungssoftware (Chef) und Application Lifecycle Management. Mithilfe des Application Lifecycle Management können Sie festlegen, wann Ressourcen eingerichtet, konfiguriert, bereitgestellt, nicht bereitgestellt oder heruntergefahren werden.

Zusätzliche Flexibilität erhalten Sie, da AWS OpsWorks Ihnen erlaubt, Ihre Anwendung in konfigurierbaren Stacks zu definieren. Sie können auch vordefinierte Anwendungs-Stacks auswählen. Anwendungs-Stacks enthalten die gesamte Bereitstellung für AWS-Ressourcen, die Ihre Anwendung benötigt, einschließlich Anwendungsserver, Webserver, Datenbanken und Load Balancer.

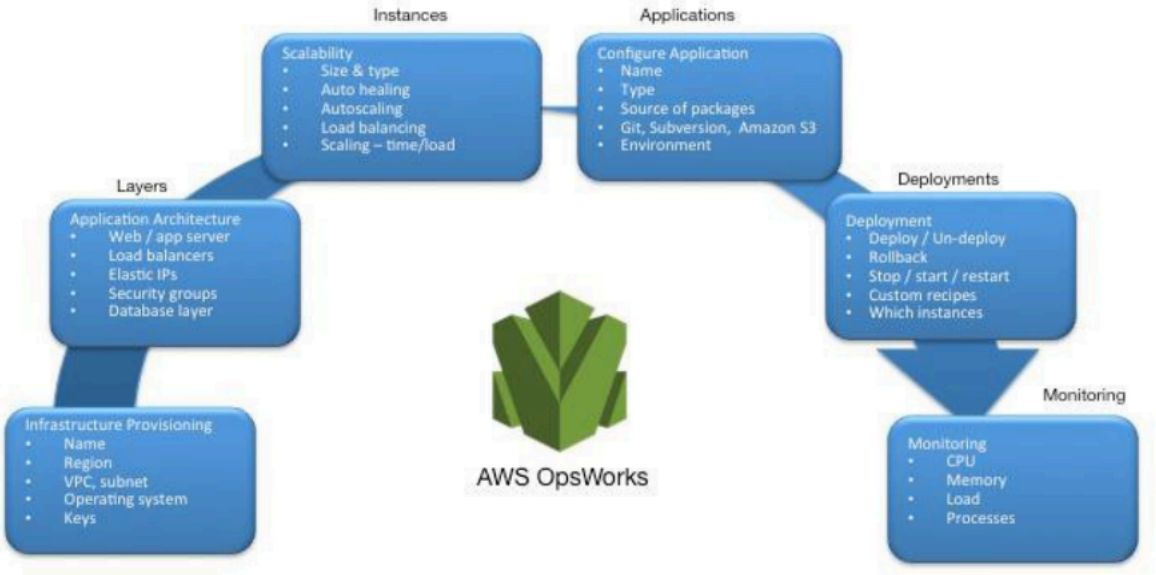

Abbildung 2 – AWS OpsWorks mit DevOps-Funktionen und -Architektur

Anwendungs-Stacks sind in architektonische Schichten unterteilt, sodass Stacks unabhängig voneinander verwaltet werden können. Schichten können beispielsweise Webstufen, Anwendungsebenen und Datenbankebenen sein. AWS OpsWorks vereinfacht auch die Einrichtung von Auto-Scaling-Gruppen und Elastic-Load-Balancing-Load-Balancer, wodurch das DevOps-Prinzip der Automatisierung weiter veranschaulicht wird. AWS OpsWorks unterstützt genau wie AWS Elastic Beanstalk Anwendungsversionen, kontinuierliche Bereitstellung und Infrastrukturkonfigurationsmanagement.

Außerdem unterstützt AWS OpsWorks die DevOps-Praktiken der Überwachung und Protokollierung (diese werden im nächsten Abschnitt behandelt). Die Überwachungsunterstützung wird von Amazon CloudWatch bereitgestellt. Alle Lebenszyklusereignisse werden protokolliert und ein separates Chef-Protokoll dokumentiert alle ausgeführten Chef-Rezepte mit allen Ausnahmen.

AWS Elastic Beanstalk

AWS Elastic Beanstalk ist ein Service für die schnelle Bereitstellung und Skalierung von Webanwendungen, die mit Java, .NET, PHP, Node.js, Python, Ruby, Go und Docker auf vertrauten Servern wie Apache, GINX, Passenger und IIS entwickelt werden.

Elastic Beanstalk ist eine Abstraktion zu Amazon EC2, Auto Scaling und vereinfacht die Bereitstellung durch zusätzliche Funktionen wie Klonen, Blau/Grün-Bereitstellungen, Elastic Beanstalk Command Line Interface (eb cli) und Integration in AWS Toolkit for Visual Studio, Visual Studio Code, Eclipse und IntelliJ zur Steigerung der Entwicklerproduktivität.

Überwachung und Protokollierung

Kommunikation und Zusammenarbeit sind auch in einer DevOps-Philosophie elementar. Für die Umsetzung ist jedoch Feedback unerlässlich. In AWS wird Feedback von zwei Kernservices bereitgestellt: Amazon CloudWatch und AWS CloudTrail. Zusammen bieten sie eine robuste Infrastruktur für Überwachung, Warnung und Prüfung, sodass Entwickler und Produktionsteams eng und transparent zusammenarbeiten können.

AWS bietet die folgenden Services zur Überwachung und Protokollierung:

Themen

Amazon CloudWatch

Amazon CloudWatch-Metriken erfassen automatisch Daten aus AWS-Services wie Amazon-EC2-Instances, Amazon-EBS-Volumes und Amazon-RDS-DB-Instances. Diese Metriken können dann in Dashboards organisiert werden. Außerdem können Alarme oder Ereignisse erstellt werden, um Ereignisse auszulösen oder Auto-Scaling-Aktionen auszuführen.

Amazon-CloudWatch-Alarme

Sie können Alarme basierend auf den von Amazon CloudWatch Metrics gesammelten Metriken einrichten. Der Alarm kann dann eine Benachrichtigung an das Amazon Simple Notification Service (Amazon SNS)-Thema senden oder Auto-Scaling-Aktionen einleiten. Ein Alarm erfordert einen Zeitraum (Zeitdauer für die Auswertung einer Metrik), einen Auswertungszeitraum (Anzahl der letzten Datenpunkte) und Datenpunkte zum Alarm (Anzahl der Datenpunkte innerhalb des Auswertungszeitraums).

Amazon CloudWatch Logs

Amazon CloudWatch Logs ist ein Service zur Protokollaggregation und -überwachung. AWS CodeBuild, CodeCommit, CodeDeploy und CodePipeline bieten Integrationen von CloudWatch-Protokollen, sodass alle Protokolle zentral überwacht werden können. Darüber hinaus ermöglichen die zuvor genannten Services sowie verschiedene andere AWS-Services eine direkte Integration von CloudWatch.

Mit CloudWatch Logs können Sie:

- Ihre Protokolldaten abfragen
- Protokolle von Amazon-EC2-Instances überwachen
- Protokollierte AWS-CloudTrail-Ereignisse überwachen
- Eine Richtlinie zur Protokollaufbewahrung festlegen

Amazon CloudWatch Logs Insights

Amazon CloudWatch Logs Insights scannt Ihre Protokolle und ermöglicht Ihnen die Durchführung interaktiver Abfragen und Visualisierungen. Der Service kann verschiedene Protokollformate verarbeiten und erkennt automatisch Felder aus JSON-Protokollen.

Amazon CloudWatch Events

Amazon CloudWatch Events bietet einen Stream von Systemereignissen nahezu in Echtzeit, der Änderungen an AWS-Ressourcen beschreibt. Mit einfachen Regeln, die sich schnell einrichten lassen, können Sie Ereignisse zuordnen und sie zu einer oder mehreren Zielfunktionen oder Streams umleiten. CloudWatch Events erkennt betriebsbezogene Veränderungen, sobald diese auftreten. Zudem reagiert CloudWatch Events auf diese betriebsbezogenen Änderungen und führt bei Bedarf Korrekturmaßnahmen durch, indem es an die Umgebung Nachrichten versendet, Funktionen aktiviert, Änderungen vornimmt und Statusinformationen erfasst.

Sie können Regeln in CloudWatch Events konfigurieren, um über Änderungen in AWS-Services benachrichtigt zu werden und diese Ereignisse mithilfe von Amazon EventBridge in andere Systeme von Drittanbietern zu integrieren. Im Folgenden sind die AWS DevOps-bezogenen Services aufgeführt, die in CloudWatch Events integriert sind.

- Auto-Scaling-Ereignisse für Anwendungen

- CodeBuild-Ereignisse
- CodeCommit-Ereignisse
- CodeDeploy-Ereignisse
- CodePipeline-Ereignisse

Amazon EventBridge

Amazon CloudWatch Events und EventBridge liegen der gleiche Service und die gleiche API zugrunde, EventBridge bietet jedoch mehr Funktionen.

Amazon EventBridge ist ein Serverless Event Bus, der Integrationen zwischen AWS-Services, Software-as-a-Service (SaaS) und Ihren Anwendungen ermöglicht. Zusätzlich zum Erstellen ereignisgesteuerter Anwendungen kann EventBridge verwendet werden, um über die Ereignisse der Services wie CodeBuild, CodeDeploy, CodePipeline und CodeCommit zu informieren.

AWS CloudTrail

Um die DevOps-Prinzipien der Zusammenarbeit, Kommunikation und Transparenz zu berücksichtigen, ist es wichtig, zu wissen, wer Änderungen an Ihrer Infrastruktur vornimmt. In AWS wird diese Transparenz vom AWS CloudTrail-Service bereitgestellt. Alle AWS-Interaktionen werden über AWS-API-Aufrufe verarbeitet, die von AWS CloudTrail überwacht und protokolliert werden. Alle generierten Protokolldateien werden in einem Amazon-S3-Bucket gespeichert, den Sie definieren. Protokolldateien werden mit serverseitiger Amazon S3-Verschlüsselung (SSE) verschlüsselt.
Alle API-Aufrufe werden protokolliert, unabhängig davon, ob sie direkt von einem Benutzer oder im Namen eines Benutzers von einem AWS-Service stammen. Zahlreiche Gruppen können von CloudTrail-Protokollen profitieren, darunter Produktionsteams für Support, Sicherheitsteams für Governance und Finanzteams für die Abrechnung.

Kommunikation und Zusammenarbeit

Egal, ob Sie die DevOps-Kultur in Ihrem Unternehmen einführen oder die DevOps-Kommunikation einem kulturellen Wandel durchläuft, Zusammenarbeit ist ein wichtiger Teil Ihres Ansatzes. Bei Amazon haben wir erkannt, dass die Denkweise der Teams geändert werden muss. Daher haben wir das Konzept der Zwei-Pizza-Teams eingeführt.

Themen

Zwei-Pizza-Teams

„Wir versuchen, Teams zu bilden, die nur so groß sind, dass sie von zwei Pizzen satt werden“, sagte Bezos. „Wir nennen das die Zwei-Pizza-Team-Regel.“

Je kleiner das Team, desto besser ist die Zusammenarbeit. Zusammenarbeit ist auch sehr wichtig, da die Software-Releases an Tempo zunehmen. Und die Fähigkeit eines Teams, die Software bereitzustellen, kann für Ihr Unternehmen ein Alleinstellungsmerkmal gegenüber Ihrer Mitbewerber sein. Stellen Sie sich eine Situation vor, in der eine neue Produktfunktion veröffentlicht oder ein Programmfehler behoben werden muss. Sie möchten, dass dies so schnell wie möglich geschieht, damit Sie eine schnellere Markteinführung vornehmen können. Dies ist auch wichtig, da die Transformation kein langsamer Prozess sein soll, sondern ein agiler Ansatz, bei dem Wellen von Veränderungen Wirkung zeigen.

Die Kommunikation zwischen den Teams ist ebenfalls wichtig: Wir bewegen uns auf das Modell der geteilten Verantwortung zu und lassen den Ansatz der isolierten Entwicklung hinter uns. Damit wird das Konzept der Eigenverantwortung in das Team eingeführt und seine Perspektive auf eine End-to-End-Betrachtungsweise verlagert. Ihr Team sollte Ihre Produktionsumgebungen nicht als Black Boxes betrachten, in denen sie keine Sichtbarkeit haben.

Der kulturelle Wandel ist auch wichtig, weil Sie möglicherweise ein gemeinsames DevOps-Team aufbauen oder der andere Ansatz darin besteht, dass Sie ein oder mehrere DevOps-fokussierte Mitglieder in Ihrem Team haben. Beide Ansätze führen zu geteilter Verantwortung im Team.

Sicherheit

Egal, ob Sie eine DevOps-Transformation durchlaufen oder DevOps-Prinzipien zum ersten Mal implementieren, Sie sollten darüber nachdenken, wie Sicherheit in Ihre DevOps-Prozesse integriert ist. Dieses Thema sollte sich durch alle Ihre Build- und Testbereitstellungsphasen ziehen.

Bevor wir über Sicherheit in DevOps in AWS sprechen, sehen wir uns das AWS-Modell der geteilten Verantwortung an.

Themen

Das AWS-Modell der geteilten Verantwortung

Sowohl AWS als auch der Kunde sind für die Sicherheit verantwortlich. Die verschiedenen Komponenten des Modells der geteilten Verantwortung werden im Folgenden erläutert:

- AWS-Verantwortung „Sicherheit der Cloud“: AWS ist für den Schutz der Infrastruktur verantwortlich, in der alle in der AWS Cloud angebotenen Services ausgeführt werden. Diese Infrastruktur umfasst die Hardware, Software, Netzwerke und Einrichtungen, in bzw. auf denen AWS-Cloud-Services ausgeführt werden.
- Kundenverantwortung „Sicherheit in der Cloud“: Die Verantwortung des Kunden wird durch die AWS-Cloud-Services bestimmt, die ein Kunde auswählt. Dadurch wird der Umfang der Konfiguration bestimmt, die der Kunde im Rahmen seiner Sicherheitsverantwortung durchführen muss.

Durch dieses gemeinsame Modell kann der Kunde entlastet werden, da AWS die Komponenten vom Host-Betriebssystem und der Virtualisierungsebene bis hin zur physischen Sicherheit der Einrichtungen, in denen der Service ausgeführt wird, betreibt, verwaltet und kontrolliert. Dies ist in Fällen, in denen Kunden die Sicherheit ihrer Build-Umgebungen verstehen möchten, von entscheidender Bedeutung.

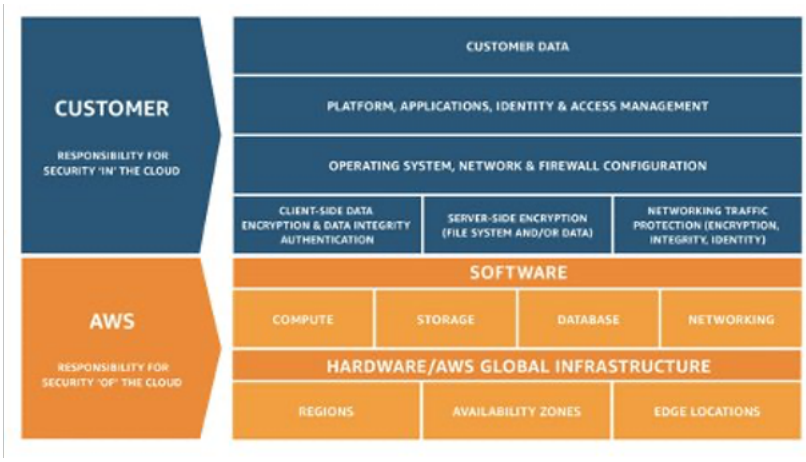

Abbildung 3: AWS-Modell der geteilten Verantwortung

Identitäts- und Zugriffsverwaltung

AWS Identity and Access Management (IAM) definiert die Kontrollen und Richtlinien, die zur Verwaltung des Zugriffs auf AWS-Ressourcen verwendet werden. Mit IAM können Sie Benutzer und Gruppen erstellen und Berechtigungen für verschiedene DevOps-Services definieren.

Zusätzlich zu den Benutzern benötigen verschiedene Services möglicherweise auch Zugriff auf AWS-Ressourcen. Beispielsweise benötigt Ihr CodeBuild-Projekt möglicherweise Zugriff zum Speichern von Docker-Images in der Amazon Elastic Container Registry (Amazon ECR) sowie Berechtigungen zum Schreiben in Amazon ECR. Diese Berechtigungstypen werden durch eine spezielle Typrolle definiert, die als Servicerolle bezeichnet wird.

IAM ist eine Komponente der AWS-Sicherheitsinfrastruktur. Mit IAM können Sie Gruppen, Benutzer, Servicerollen und Anmeldeinformationen wie Passwörter, Zugriffsschlüssel und Berechtigungsrichtlinien für den Zugriff auf AWS-Services und -Ressourcen zentral verwalten. Mit einer IAM-Richtlinie können Sie den Berechtigungssatz definieren. Diese Richtlinie kann dann entweder an eine Rolle, einen Benutzer oder einen Service angefügt werden, um die entsprechende Berechtigung zu definieren. Sie können IAM auch verwenden, um Rollen zu erstellen, die in Ihrer gewünschten DevOps-Strategie weit verbreitet sind. In einigen Fällen kann es durchaus sinnvoll sein, programmgesteuert AssumeRole zu übernehmen, anstatt die Berechtigungen direkt abzurufen. Wenn ein Service oder Benutzer Rollen übernimmt, erhält er temporäre Anmeldeinformationen für den Zugriff auf einen Service, auf den er normalerweise nicht zugreifen kann.

Fazit

Um den Weg in die Cloud reibungslos, effizient und effektiv zu gestalten, sollten Technologieunternehmen die DevOps-Prinzipien und -Praktiken anwenden. Diese Prinzipien sind in die AWS-Plattform eingebettet. In der Tat bilden sie den Eckpfeiler zahlreicher AWS-Services, insbesondere der Bereitstellungs- und Überwachungsangebote.

Definieren Sie zunächst Ihre Infrastruktur mit dem Service AWS CloudFormation oder AWS Cloud Development Kit (AWS CDK). Als Nächstes legen Sie mithilfe von Services wie AWS CodeBuild, AWS CodeDeploy, AWS CodePipeline und AWS CodeCommit fest, wie Ihre Anwendungen die kontinuierliche Bereitstellung nutzen werden. Verwenden Sie auf Anwendungsebene Container wie AWS Elastic Beanstalk, Amazon Elastic Container Service (Amazon ECS) oder Amazon Elastic Kubernetes Service (Amazon EKS) und AWS OpsWorks, um die Konfiguration gängiger Architekturen zu vereinfachen. Durch Verwendung dieser Services lassen sich auch andere wichtige Services wie Auto Scaling und Elastic Load Balancing leicht einbeziehen. Verwenden Sie abschließend die DevOps-Überwachungsstrategie wie Amazon CloudWatch und solide Sicherheitsmethoden wie AWS IAM.

Mit AWS als Partner sorgen Ihre DevOps-Prinzipien für Agilität in Ihrem Unternehmen und Ihrer IT-Organisation und beschleunigen Ihren Weg in die Cloud.

Dokumentversionen

Abonnieren Sie den RSS-Feed, um über Aktualisierungen dieses Whitepapers benachrichtigt zu werden.

Update-Historie-Änderung	Update-Historie-Beschreibung	Update-Historie-Datum
Fehlender Abschnitt „Mitwirkende“ wiederhergestellt	Der fehlende Abschnitt „Mitwirkende“ wurde wiederhergestellt und kleinere Textänderungen wurden vorgenommen.	21. November 2020
Abschnitte wurden aktualisiert, um neue Services zu berücksichtigen.	Abschnitte wurden aktualisiert, um neue Services zu berücksichtigen.	16. Oktober 2020
Erstveröffentlichung	Erstveröffentlichung des Whitepapers	1. Dezember 2014

Mitwirkende

An diesem Dokument haben folgende Personen mitgewirkt:

- Muhammad Mansoor, Solutions Architect
- Ajit Zadgaonkar, World Wide Tech Leader, Modernization
- Juan Lamadrid – Solutions Architect
- Darren Ball – Solutions Architect
- Rajeswari Malladi – Solutions Architect
- Pallavi Nargund – Solutions Architect
- Bert Zahniser – Solutions Architect
- Abdullahi Olaoye – Cloud Solutions Architect
- Mohamed Kiswani – Software Development Manager
- Tara McCann – Manager Solutions Architect

Hinweise

Kunden sind eigenverantwortlich für die unabhängige Bewertung der Informationen in diesem Dokument zuständig. Dieses Dokument: (a) dient rein zu Informationszwecken, (b) spiegelt die aktuellen Produktangebote und Verfahren von AWS wider, die sich ohne vorherige Mitteilung ändern können, und (c) impliziert keinerlei Verpflichtungen oder Zusicherungen seitens AWS und dessen Tochtergesellschaften, Lieferanten oder Lizenzgebern. AWS-Produkte oder -Services werden im vorliegenden Zustand und ohne ausdrückliche oder stillschweigende Gewährleistungen, Zusicherungen oder Bedingungen bereitgestellt. Die Verantwortung und Haftung von AWS gegenüber seinen Kunden wird durch AWS-Vereinbarungen geregelt. Dieses Dokument ist weder ganz noch teilweise Teil der Vereinbarungen zwischen AWS und seinen Kunden und ändert diese Vereinbarungen auch nicht.