Auswahl des richtigen GitOps Tools für Ihren Amazon EKS-Cluster

AWS Präskriptive Leitlinien

AWS Präskriptive Leitlinien: Auswahl des richtigen GitOps Tools für Ihren Amazon EKS-Cluster

Table of Contents

Auswahl des richtigen GitOps Tools für Ihren Amazon EKS-Cluster

Pradip Kumar Pandey und Pratap Kumar Nanda, Amazon Web Services ()AWS

April 2025 (Geschichte der Dokumente)

In der sich schnell entwickelnden Landschaft cloudnativer Technologien GitOps hat sich diese Methode zu einer leistungsstarken Methode für die Verwaltung und Bereitstellung von Anwendungen und Infrastruktur entwickelt. Wenn Sie Amazon Elastic Kubernetes Service (Amazon EKS) verwenden, können GitOps Implementierungsprinzipien Ihre Bereitstellungsprozesse erheblich verbessern, die Zuverlässigkeit verbessern und den Betrieb rationalisieren. Es stehen eine Vielzahl von GitOps Tools zur Verfügung, und die Auswahl des richtigen Tools für Ihren EKS-Cluster ist eine wichtige Entscheidung, die sich auf die Effizienz Ihres Teams und den Gesamterfolg Ihrer Verfahren auswirken kann. DevOps

Bei der Auswahl eines geeigneten GitOps Tools für Ihre Amazon EKS-Umgebung müssen verschiedene Faktoren sorgfältig berücksichtigt werden, darunter Ihre spezifischen Anforderungen, das Fachwissen Ihres Teams, die Skalierbarkeitsanforderungen und die Integrationsmöglichkeiten mit bestehenden Tools AWS-Services. Jedes Tool hat seine eigenen Funktionen, Stärken und potenziellen Einschränkungen. Daher ist es wichtig, dass Sie Ihre Wahl an den Zielen und dem betrieblichen Kontext Ihres Unternehmens ausrichten.

Dieser Leitfaden untersucht die wichtigsten Überlegungen bei der Auswahl von GitOps Tools für Amazon EKS, vergleicht häufig verwendete Optionen und bietet Einblicke, die Ihnen helfen, eine fundierte Entscheidung zu treffen. Es behandelt neun beliebte GitOps Tools:

- Argo-CD
- Flux
- Weben GitOps
- Jenkins X
- GitLab CI/CD
- Spinnaker
- Rancher-Flotte
- Codefresh
- Pulumi

Gezielte Geschäftsergebnisse

In der folgenden Liste werden mögliche Ziele und Ergebnisse beschrieben, wenn Sie sich für ein Tool zur Implementierung von GitOps Prinzipien in Ihren Entwicklungs- und Betriebsprozessen entscheiden.

Nahtlose Integration mit Amazon EKS

Ihr GitOps Tool sollte sich problemlos in Amazon EKS integrieren lassen und Kompatibilität mit Amazon EKS-spezifischen Funktionen und Optimierungen bieten.

- Native Amazon EKS-Unterstützung: Suchen Sie nach Tools, die integrierte Unterstützung für Amazon EKS bieten, einschließlich einfacher Clusterverbindung und -verwaltung.
- AWS-Service Integration: Stellen Sie sicher, dass das Tool mit anderen Tools AWS-Services wie AWS Identity and Access Management (IAM), Amazon Elastic Container Registry (Amazon ECR) und Amazon interagieren kann. CloudWatch
- Amazon EKS-Add-On-Kompatibilität: Vergewissern Sie sich, dass das Tool Amazon EKS-Add-Ons unterstützt und diese effektiv verwalten kann.

Skalierbarkeit und Leistung

Ihr GitOps Tool sollte in der Lage sein, den Umfang Ihrer Amazon EKS-Operationen zu bewältigen, von kleinen Clustern bis hin zu großen Umgebungen mit mehreren Clustern.

- Ressourceneffizienz: Bewerten Sie den Ressourcenverbrauch des Tools und seine Auswirkungen auf die Cluster-Leistung.
- Umfangreicher Betrieb: Beurteilen Sie, ob das Tool in der Lage ist, zahlreiche Anwendungen und Cluster gleichzeitig zu verwalten.
- Leistung unter Last: Überlegen Sie, wie das Tool bei häufig aktualisierten Updates und umfangreichen Bereitstellungen abschneidet.

Sicherheits und Compliance

Sicherheitsfunktionen und Compliance-Funktionen sind von entscheidender Bedeutung, insbesondere in regulierten Branchen oder beim Umgang mit sensiblen Daten.

- Zugriffskontrolle: Halten Sie Ausschau nach robusten Funktionen für die rollenbasierte Zugriffskontrolle (RBAC), die sich in IAM integrieren lassen.
- Verwaltung von Geheimnissen: Evaluieren Sie, wie das Tool mit vertraulichen Informationen umgeht und sich in unsere anderen Lösungen integrieren lässt. AWS Secrets Manager
- Prüfprotokolle: Stellen Sie sicher, dass das Tool umfassende Protokollierungs- und Prüffunktionen zur Einhaltung von Vorschriften und zur Fehlerbehebung bietet.
- Sicherheitsscans: Ziehen Sie Tools in Betracht, die integrierte Sicherheitsscans nach Sicherheitslücken in Bereitstellungen bieten.

Benutzerfreundlichkeit und Lernkurve

Das Tool sollte benutzerfreundlich sein und den Fähigkeiten Ihres Teams entsprechen, um eine schnelle Einführung und effiziente Nutzung zu gewährleisten.

- Benutzeroberfläche: Beurteilen Sie die Intuitivität der Funktionen der Befehlszeilenschnittstelle (CLI) und der grafischen Benutzeroberfläche (GUI).
- Qualität der Dokumentation: Halten Sie Ausschau nach umfassenden up-to-date Dokumentationen und Tutorials.
- Lernressourcen: Berücksichtigen Sie die Verfügbarkeit von Schulungsmaterialien, Kursen und Community-Ressourcen.

Community- und Netzwerkunterstützung

Eine starke Community und ein starkes Netzwerk können wertvolle Ressourcen, Plugins und langfristige Nachhaltigkeit bieten.

- Aktive Entwicklung: Überprüfe die Häufigkeit der Updates und die Reaktionsfähigkeit der Betreuer.
- Größe der Community: Berücksichtigen Sie die Größe und Aktivität der Benutzergemeinschaft für Support und Wissensaustausch.
- Integrationen von Drittanbietern: Evaluieren Sie die Verfügbarkeit von Plugins und Integrationen mit anderen Tools in Ihrem Stack.

Verwaltungsfunktionen für mehrere Cluster

Wenn Sie über mehrere EKS-Cluster verfügen, ist die Fähigkeit, diese effizient zu verwalten, von entscheidender Bedeutung.

- Zentralisiertes Management: Suchen Sie nach Funktionen, mit denen Sie mehrere Cluster von einer einzigen Steuerungsebene aus verwalten können.
- Clusterverbund: Ziehen Sie Tools in Betracht, die den Kubernetes-Verbund für Multi-Cluster-Anwendungen unterstützen.
- Umgebungsparität: Beurteilen Sie, wie gut das Tool die Konsistenz in verschiedenen Umgebungen wie Entwicklung, Staging und Produktion gewährleistet.

Beobachtbarkeit und Überwachung

Das Tool sollte klare Einblicke in den Status Ihrer Bereitstellungen und den Zustand Ihres Clusters bieten.

- Sichtbarkeit der Bereitstellung: Suchen Sie nach Funktionen, die einen klaren Überblick über den Status und den Verlauf der Bereitstellung bieten.
- Integration mit Überwachungstools: Überlegen Sie, wie gut sich das Tool in beliebte Überwachungslösungen wie Prometheus und Grafana integrieren lässt.
- Warnfunktionen: Beurteilen Sie, ob das Tool in der Lage ist, Warnmeldungen für Bereitstellungsprobleme oder Abweichungen einzurichten und zu verwalten.

Flexibilität und Anpassung

Die Fähigkeit, das Tool an Ihre spezifischen Arbeitsabläufe und Anforderungen anzupassen, ist wichtig für die langfristige Zufriedenheit.

- Erweiterbarkeit: Suchen Sie nach Plugin-Architekturen oder nach Plugin-Architekturen APIs , mit denen Sie die Funktionalität des Tools erweitern können.
- Unterstützung für benutzerdefinierte Ressourcen: Stellen Sie sicher, dass das Tool benutzerdefinierte Kubernetes-Ressourcen effektiv verarbeiten kann.
- Workflow-Anpassung: Beurteilen Sie, wie einfach Sie die GitOps Workflows an die Bedürfnisse Ihres Teams anpassen können.

Kontinuierliche Bereitstellung und Unterstützung bei der schrittweisen Bereitstellung

Fortschrittliche Bereitstellungsstrategien sind oft entscheidend, um Risiken zu minimieren und reibungslose Updates zu gewährleisten.

- Bereitstellungen auf Canary: Halten Sie Ausschau nach integrierter Unterstützung für Canary-Versionen.
- Blue/green deployments: Assess the tool's capabilities for blue/greenBereitstellungsstrategien.
- Rollback-Mechanismen: Sorgen Sie für robuste easy-to-use Rollback-Funktionen für eine schnelle Wiederherstellung nach fehlgeschlagenen Bereitstellungen.

Kosteneffektivität und Ressourcennutzung

Berücksichtigen Sie die Gesamtkosten für die Einführung und Wartung des Tools, einschließlich der direkten und indirekten Kosten.

- Lizenzkosten: Vergleichen Sie Open-Source-Optionen mit kommerziellen Lösungen und berücksichtigen Sie Support- und Unternehmensfunktionen.
- Betriebskosten: Beurteilen Sie die zusätzlichen Betriebskosten in Bezug auf Verwaltung und Wartung.
- Ressourcenverbrauch: Bewerten Sie die Effizienz des Tools in Bezug auf die benötigten Rechen- und Speicherressourcen.

Wenn Sie diese Ergebnisse und ihre Aspekte sorgfältig abwägen, können Sie eine fundierte Entscheidung über das am besten geeignete GitOps Tool für Ihren EKS-Cluster treffen und sicherstellen, dass das Tool den Bedürfnissen, Fähigkeiten und der langfristigen Strategie Ihres Unternehmens entspricht.

GitOps Tools für EKS-Cluster

Es gibt mehrere GitOps Tools für Kubernetes, die derzeit auf dem Markt erhältlich sind. Hier ist eine Liste der am häufigsten verwendeten Optionen:

- Argo-CD
- Flux
- Weben GitOps
- Jenkins X
- GitLab CI/CD
- Spinnaker
- Rancher-Flotte
- Codefresh
- Pulumi

Folgen Sie den Links, um detaillierte Informationen darüber zu erhalten, wie diese Tools GitOps Praktiken implementieren. Jedes Tool hat Stärken und Anwendungsfälle. Die Wahl hängt von Faktoren wie Ihren spezifischen Anforderungen, der vorhandenen Infrastruktur, dem Fachwissen des Teams und den gewünschten Funktionen ab. Es ist wichtig, dass Sie diese Tools auf der Grundlage der Bedürfnisse Ihres Unternehmens und der Komplexität Ihrer Kubernetes-Umgebung bewerten.

Argo CD

Argo CD ist ein weit verbreitetes GitOps Continuous Delivery (CD) -Tool für Kubernetes, das mehreren wichtigen Prinzipien entspricht. GitOps

GitOps Unterstützung

Flächen	Funktionen des Tools
Deklarative Konfiguration	Argo CD verwendet deklarative Konfigura tionen, die in Git-Repositorys gespeichert sind. Der gewünschte Status der Anwendung und Infrastruktur ist in YAML-Dateien definiert

Flächen	Funktionen des Tools
	. Diese Konfigurationen beschreiben, was bereitgestellt werden sollte, nicht, wie sie bereitgestellt werden sollen.
Versionskontrollsystem als zentrale Informationsquelle	Git-Repositorien dienen als zentrale Informationsquelle für das gesamte System. Alle Änderungen an der Anwendung und Infrastruktur werden über Git vorgenommen. Dies gewährleistet einen vollständigen Prüfpfad und die Möglichkeit, zu einem beliebigen vorherigen Status zurückzukehren.
Automatisierte Synchronisation	Argo CD überwacht das Git-Repository kontinuierlich auf Änderungen. Wenn Änderungen erkannt werden, synchronisiert es automatisch den aktuellen Status des Clusters mit dem gewünschten Status, der in Git definiert ist. Dadurch wird sichergestellt, dass der Cluster immer den Status widerspiegelt, der im Repository beschrieben ist.
Kubernetes-nativ	Argo CD wurde speziell für Kubernetes-Umgebungen entwickelt. Es nutzt den deklarativen Charakter und die benutzerdefinierten Ressourcen in Kubernetes für die Verwaltung von Anwendungen.
Selbstheilung und Drift-Erkennung	Argo CD vergleicht regelmäßig den Live-Status des Clusters mit dem gewünschten Status in Git. Wenn es Abweichungen (Unterschiede zwischen dem tatsächlichen und dem gewünschten Zustand) feststellt, kann es diese Abweichungen automatisch korrigieren.

Flächen	Funktionen des Tools
Unterstützung für mehrere Cluster und mehrere Mandanten	Argo CD kann mehrere Kubernetes-Cluster von einer einzigen Instanz aus verwalten. Es unterstützt Mehrmandantenfähigkeit, sodass verschiedene Teams ihre Anwendungen unabhängig voneinander verwalten können.
Definition der Anwendung	Anwendungen in Argo CD werden mithilfe der Anwendungs-CRD (benutzerdefinierte Ressourcendefinition) definiert. Dies ermöglich t eine Kubernetes-native Methode, um zu definieren, was und wie bereitgestellt werden soll.
Trennung von Bereitstellung und Veröffent lichung	Argo CD trennt die Bereitstellung von Code von seiner Veröffentlichung für Benutzer. Dies wird durch verschiedene Bereitstellungsstrategien erreicht, z. B. blue/green durch kanarische Bereitstellungen.
Beobachtbarkeit und Überprüfbarkeit	Argo CD bietet eine Web-Benutzeroberfläche und CLI zur Beobachtung des Status von Anwendungen und Clustern. Alle Aktionen werden protokolliert, um einen klaren Prüfpfad für Änderungen und Implementierungen zu gewährleisten.
Sicherheit und RBAC	Argo CD ist in die rollenbasierte Zugriffsk ontrolle (RBAC) von Kubernetes integriert. Es unterstützt die Single-Sign-On-Integration für Authentifizierung und Autorisierung.
Steckbare Architektur	Argo CD unterstützt verschiedene Quellcode verwaltungssysteme, Helm-Charts, Kustomize und andere Kubernetes-Manifestformate. Diese Flexibilität ermöglicht es, sich in verschiedene Umgebungen und Workflows einzufügen.

Flächen	Funktionen des Tools
Kontinuierliche Bereitstellung (CD)	Obwohl sich Argo CD auf Continuous Delivery konzentriert, kann es in CI-Tools (Continuous Integration) integriert werden, um eine vollständ ige CI/CD Pipeline zu erstellen.

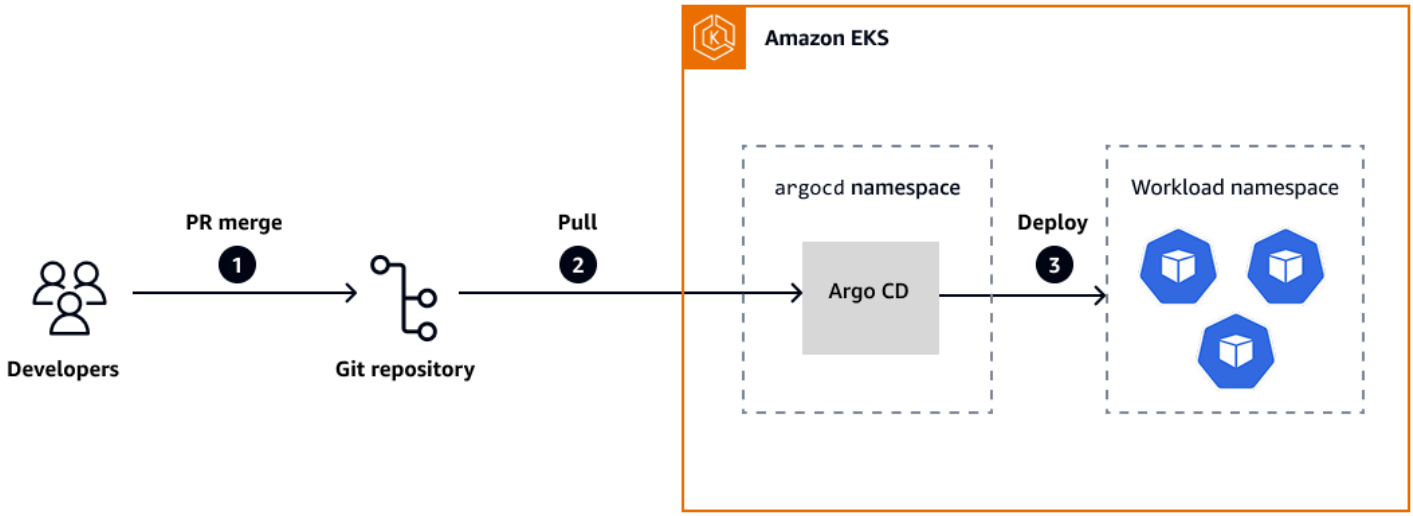
Durch die Einhaltung dieser GitOps Prinzipien bietet Argo CD eine robuste, skalierbare und sichere Methode zur Verwaltung von Kubernetes-Bereitstellungen. Es stellt sicher, dass der Betriebsstatus Ihres Systems immer mit dem gewünschten Status synchronisiert ist, der in Ihrem Git-Repository definiert ist, und sorgt für Konsistenz, Zuverlässigkeit und einfache Verwaltung in komplexen Kubernetes-Umgebungen.

Szenarien und Anforderungen, die Argo CD erfüllen kann, finden Sie weiter unten in diesem Handbuch unter Anwendungsfälle für Argo CD. Einen Vergleich zwischen Argo CD und Flux finden Sie weiter unten in diesem Handbuch unter Funktionsvergleich.

Weitere Informationen finden Sie in der Argo-CD-Dokumentation.

Architektur

Das folgende Diagramm veranschaulicht einen CD-Workflow, GitOps der Argo CD innerhalb eines EKS-Clusters verwendet. Ausführliche Informationen finden Sie in der Argo-CD-Dokumentation.



Wobei:

- Schritt 1: Zusammenführung von Pull-Requests (PR). Ein Entwickler überträgt Änderungen an Kubernetes-Manifesten oder Helm-Diagrammen, die in einem Git-Repository gespeichert sind. Wenn der PR überprüft und mit dem Hauptzweig zusammengeführt wurde, wird der gewünschte Status der Anwendung in der Quellcodeverwaltung aktualisiert.
- Schritt 2: Repository-Synchronisierung. Argo CD läuft in einem dedizierten Namespace (`argocd`) im EKS-Cluster und überwacht kontinuierlich das konfigurierte Git-Repository. Wenn es Änderungen erkennt, ruft es die neuesten Updates ab, um den deklarierten Status abzugleichen.
- Schritt 3: Bereitstellung im Ziel-Namespace. Argo CD vergleicht den gewünschten Status von Git mit dem Live-Status im Cluster. Anschließend werden die erforderlichen Änderungen auf den Ziel-Workload-Namespace angewendet, sodass die Anwendung entsprechend bereitgestellt oder aktualisiert wird. Dazu gehört die Verwaltung von Kubernetes-Ressourcen wie Deployments, Services und Secrets ConfigMaps, um die Cluster-Konsistenz mit der Git Source of Truth aufrechtzuerhalten.

Flux

Flux ist ein weiteres Tool für Kubernetes, das GitOps Prinzipien auf einzigartige Weise implementiert.

GitOps Unterstützung

Flächen	Funktionen des Tools
Git als zentrale Informationsquelle	Flux verwendet Git-Repositorys als endgültig e Quelle für die Definition des gewünschten Systemstatus. Die gesamte Konfiguration für Anwendungen und Infrastruktur wird in Git gespeichert.
Deklarative Konfiguration	Flux arbeitet mit deklarativen Beschreibungen des gewünschten Zustands Ihres Clusters. Bei diesen Beschreibungen handelt es sich in der Regel um Kubernetes-Manifeste, Helm-Diag ramme oder Kustomize-Overlays.
Automatisierte Synchronisation	Flux überwacht das Git-Repository kontinuie rlich auf Änderungen. Wenn es Änderungen

Flächen	Funktionen des Tools
	erkennt, wendet es sie automatisch auf den Cluster an.
Kubernetes-nativ	Flux besteht aus einer Reihe von Kubernetes-Controllern und benutzerdefinierten Ressourcen. Es nutzt die Erweiterungsmechanismen in Kubernetes, um Funktionen bereitzustellen. GitOps
Pull-basiertes Bereitstellungsmodell	Im Gegensatz zu herkömmlichen Push-basierten CI/CD Systemen verwendet Flux ein Pull-basiertes Modell. Der Cluster ruft den gewünschten Status aus Git ab, anstatt ein externes System zu verwenden, um Änderungen zu übertragen.
Kontinuierlicher Abgleich	Flux vergleicht ständig den aktuellen Zustand des Clusters mit dem gewünschten Zustand in Git. Es korrigiert automatisch jede Abweichung, die zwischen diesen Zuständen erkannt wird.
Mehrmandantenfähigkeit	Flux unterstützt Mehrmandantenfähigkeit durch seine Konzepte für Anpassungen und. HelmReleases Verschiedene Teams können ihre eigenen Teile der Konfiguration unabhängig voneinander verwalten.
Progressive Lieferung	Flux unterstützt mit seiner Flagger-Komponente fortschrittliche Bereitstellungsstrategien wie Veröffentlichungen und A/B Tests auf Canary.
Helmintegration	Flux bietet native Unterstützung für Helm, sodass Sie Helm-Versionen einfach verwalten können GitOps.

Flächen	Funktionen des Tools
Automatisierung der Image-Aktualisierung	Flux kann Container-Images in Git automatisch aktualisieren, wenn neue Versionen in der Container-Registry verfügbar sind.
Unterstützung anpassen	Sie können die native Unterstützung von Flux für Kustomize verwenden, um Kubernetes-Manifeste anzupassen und zu patchen.
Sicherheit und RBAC	Flux lässt sich zur Zugriffskontrolle in Kubernetes RBAC integrieren. Es unterstützt die Verwaltung von Geheimnissen über verschiedene Backends.
Beobachtbarkeit	Flux bietet Statusinformationen und Kennzahlen zu Abstimmungen und Vorgängen. Es lässt sich in Überwachungstools integrieren, um die Beobachtbarkeit zu verbessern.
Ereignisgesteuerte Architektur	Flux verwendet einen ereignisgesteuerten Ansatz zur Implementierung von Abstimmungen und Aktualisierungen.
Erweiterbarkeit	Das Tool ist so konzipiert, dass es erweiterbar ist, sodass Sie benutzerdefinierte Controller und Ressourcen hinzufügen können.
Clusterübergreifende Synchronisation	Flux unterstützt die Verwaltung mehrerer Cluster aus einem einzigen Satz von Repositorys.
Abhängigkeitsmanagement	Es ermöglicht die Definition von Abhängigkeiten zwischen verschiedenen Teilen Ihres Systems und stellt die korrekte Reihenfolge der Operationen sicher.

Flächen	Funktionen des Tools
Webhook-Empfänger	Sie können Flux so konfigurieren, dass es Webhooks von Git-Anbietern oder anderen Systemen empfängt, um den sofortigen Abgleich zu starten.

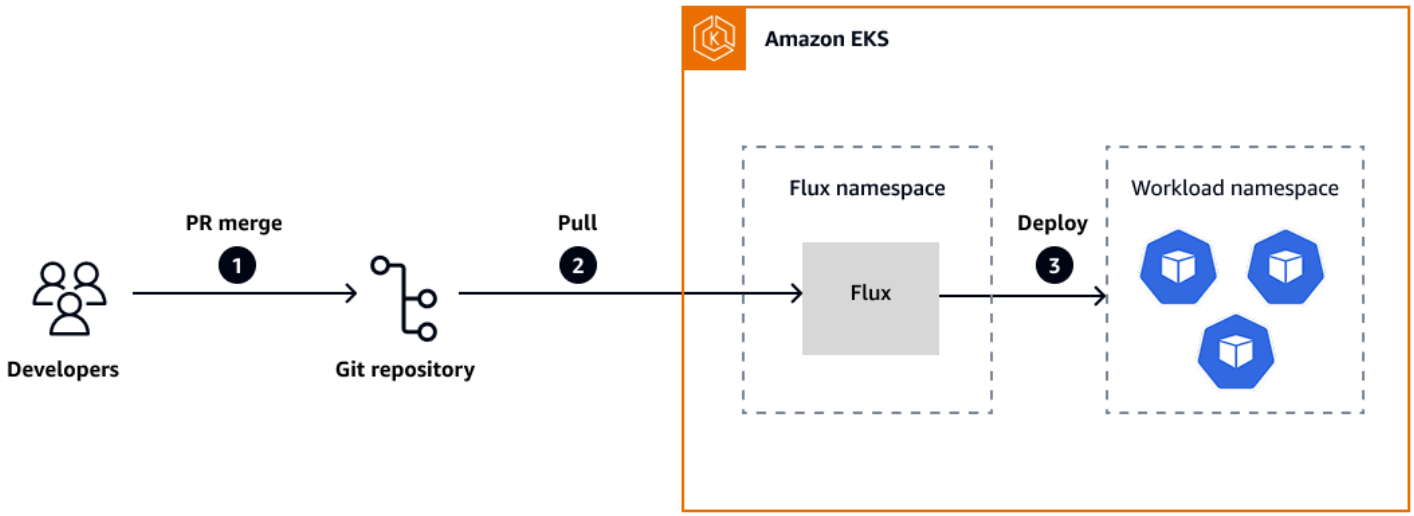
Durch die Implementierung dieser GitOps Prinzipien bietet Flux ein robustes und flexibles System für die Verwaltung von Kubernetes-Clustern und -Anwendungen. Es stellt sicher, dass Ihre Infrastruktur und Anwendungen immer mit Ihren Git-Repositorys synchronisiert sind, und sorgt für Konsistenz, Zuverlässigkeit und einfache Verwaltung in komplexen Kubernetes-Umgebungen. Durch den Kubernetes-nativen Ansatz und den Fokus auf Automatisierung eignet sich das Tool besonders gut für Cloud-native Umgebungen.

Szenarien und Anforderungen, die Flux erfüllen kann, finden Sie weiter unten in diesem Handbuch unter Flux-Anwendungsfälle. Einen Vergleich zwischen Argo CD und Flux finden Sie weiter unten in diesem Handbuch unter Funktionsvergleich.

Weitere Informationen finden Sie in der Flux-Dokumentation.

Architektur

Das folgende Diagramm zeigt einen GitOps CD-gesteuerten Workflow, der Flux innerhalb eines EKS-Clusters verwendet. Detaillierte Informationen finden Sie in der Flux-Dokumentation.



Wobei:

- Schritt 1: Zusammenführung von Pull-Requests (PR). Ein Entwickler überträgt Änderungen an Kubernetes-Manifesten oder Helm-Diagrammen, die in einem Git-Repository gespeichert sind. Wenn der PR überprüft und mit dem Hauptzweig zusammengeführt wurde, wird der gewünschte Status der Anwendung in der Quellcodeverwaltung aktualisiert.
- Schritt 2: Repository-Synchronisierung. Flux läuft in einem dedizierten Namespace im EKS-Cluster und überwacht kontinuierlich das konfigurierte Git-Repository. Wenn es Änderungen erkennt, ruft es die neuesten Updates ab, um den deklarierten Status abzugleichen.
- Schritt 3: Bereitstellung im Ziel-Namespace. Flux vergleicht den gewünschten Status von Git mit dem Live-Status im Cluster. Anschließend werden die erforderlichen Änderungen auf den Ziel-Workload-Namespace angewendet, sodass die Anwendung entsprechend bereitgestellt oder aktualisiert wird.

Weben GitOps

Weave GitOps wurde von Weaveworks entwickelt, dem Unternehmen, das den Begriff eingeführt hat. GitOps Dieses Tool bietet eine umfassende GitOps Lösung, die auf den Kernprinzipien aufbaut. GitOps

GitOps Unterstützung

Flächen	Funktionen des Tools
Git als zentrale Informationsquelle	Weave GitOps verwendet Git-Repositorys als maßgebliche Quelle für die Definition des gewünschten Systemstatus. Alle Konfigura tionen, einschließlich Anwendungsmanifeste, Infrastrukturdefinitionen und Richtlinien, werden in Git gespeichert.
Deklarative Konfiguration	Das System stützt sich auf deklarative Beschreibungen des gesamten Systemstatus. Bei diesen Beschreibungen handelt es sich in der Regel um Kubernetes-Manifeste, Helm-Diagramme oder andere deklarative Formate.
Automatisierte Synchronisation	Weave überwacht Git-Repositorys GitOps kontinuierlich auf Änderungen. Wenn es

Flächen	Funktionen des Tools
	Änderungen erkennt, wendet es sie automatis ch auf die Zielumgebung an.
Kubernetes-native Architektur	Weave GitOps besteht aus einer Reihe von Kubernetes-Controllern und benutzerdefinierten Ressourcen. Es nutzt die Erweiterungsmechan ismen in Kubernetes, um Funktionen bereitzus tellen. GitOps
Kontinuierliche Versöhnung	Dieses Tool vergleicht ständig den aktuellen Status des Clusters mit dem gewünschten Status, der in Git definiert ist. Es korrigiert automatisch jede Abweichung, die zwischen diesen Zuständen festgestellt wird.
Verwaltung mehrerer Cluster	Weave GitOps unterstützt die Verwaltun g mehrerer Kubernetes-Cluster von einer einzigen Steuerungsebene aus. Es ermöglicht eine konsistente Anwendungsbereitstellung in verschiedenen Umgebungen.
Richtlinie als Code	Weave GitOps beinhaltet das Konzept der Richtlinie als Code zur Durchsetzung von Sicherheits- und Compliance-Regeln. Richtlini en werden zusammen mit Anwendungscode und Infrastrukturdefinitionen versionsgesteuert.
Progressive Bereitstellung	Dieses Tool unterstützt fortgeschrittene Bereitstellungsstrategien wie Veröffent lichungen und blue/green Bereitstellungen auf Canary. Es ist in Flagger integriert und ermöglicht so eine automatisierte, progressive Bereitstellung.

Flächen	Funktionen des Tools
Beobachtbarkeit und Dashboards	Weave GitOps bietet integrierte Dashboards zur Überwachung des Status von Anwendungen und Clustern. Es bietet Einblicke in Abstimmungsprozesse und den Zustand von Clustern.
Von Haus aus sicher	Das Tool implementiert bewährte Sicherheitsverfahren, einschließlich RBAC-Integration und Verwaltung von Geheimnissen. Es unterstützt verschiedene Authentifizierungsmethoden und lässt sich in Identitätsanbieter für Unternehmen integrieren.
Erweiterbarkeit und Integration	Das Tool ist so konzipiert, dass es mit einer Vielzahl von Cloud-nativen Tools funktioniert. Es unterstützt beliebte Tools wie Flux, Helm und Kustomize.
Self-Service-Entwicklerplattformen	Weave GitOps ermöglicht die Schaffung von Self-Service-Plattformen für Entwickler. Es bietet Vorlagen und Leitplanken für die Anwendungsbereitstellung.
GitOps -Automatisierung	Das Tool automatisiert viele Aspekte des GitOps Workflows, einschließlich der Generierung von Pull-Requests für Updates.
Pipelines für die kontinuierliche Lieferung	Es lässt sich in CI/CD Systeme integrieren, um end-to-end Lieferleitungen zu erstellen.
Überwachung und Compliance	Weave DevOps bietet einen vollständigen Prüfpfad aller Änderungen und Aktionen. Es hilft Ihnen, Compliance-Anforderungen durch Versionskontrolle und automatisierte Prozesse zu erfüllen.

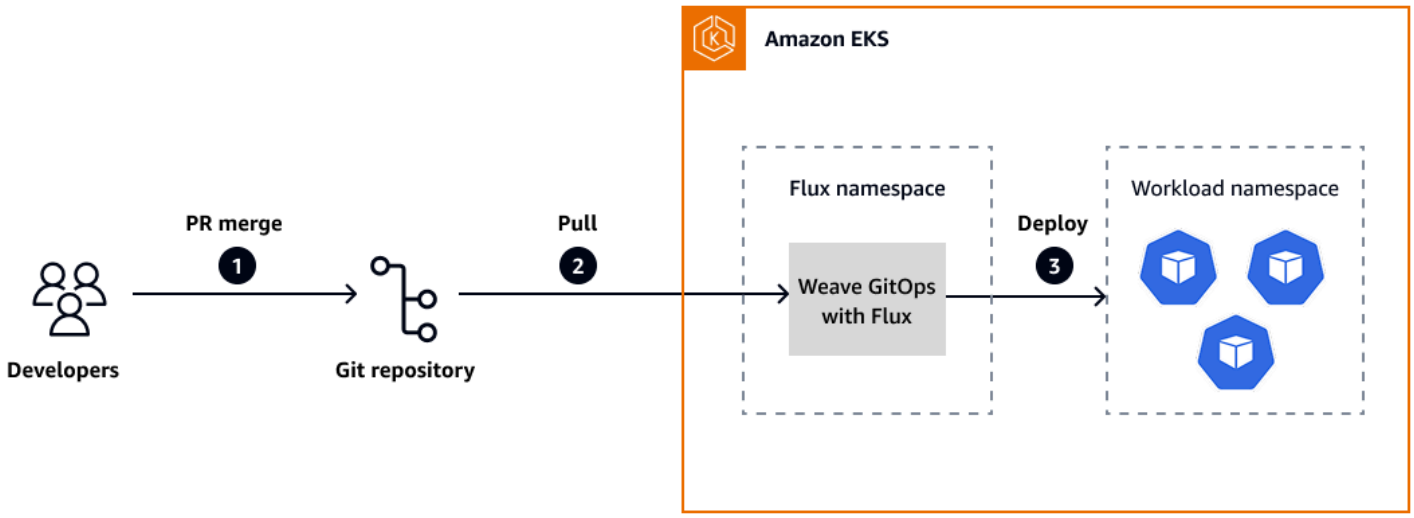
Flächen	Funktionen des Tools
Skalierbarkeit	Das Tool ist so konzipiert, dass es von kleinen Projekten bis hin zu großen Implementierungen auf Unternehmensebene skaliert werden kann.
Zusammenarbeit im Team	Weave GitOps erleichtert die Zusammenarbeit zwischen Entwicklungs- und Betriebsteams durch Git-basierte Workflows.
GitOps als Service	Dieses Tool wird GitOps als verwalteter Service angeboten, was die Einführung und Verwaltung vereinfacht.
Hybrid- und Multi-Cloud-Unterstützung	Weave GitOps ermöglicht ein konsistentes Management für verschiedene Cloud-Anbieter und lokale Umgebungen.
Kontinuierliche Sicherheit	Das Tool integriert Sicherheitsscans und die Durchsetzung von Richtlinien während des gesamten Bereitstellungsprozesses.

Weave GitOps implementiert diese Prinzipien, um eine umfassende GitOps Lösung bereitzustellen, die über die grundlegende Automatisierung der Bereitstellung hinausgeht. Ziel ist es, ein vollständiges Betriebsmodell für Cloud-native Anwendungen zu entwickeln, das sich auf Sicherheit, Skalierbarkeit und Benutzerfreundlichkeit konzentriert. Durch die Einhaltung dieser GitOps Prinzipien GitOps hilft Weave Unternehmen dabei, eine konsistente, überprüfbare und effiziente Verwaltung ihrer Kubernetes-Umgebungen über mehrere Cluster und Cloud-Anbieter hinweg zu erreichen.

Weitere Informationen finden Sie in der Weave-Dokumentation. GitOps

Architektur

Das folgende Diagramm veranschaulicht einen CD-Workflow, GitOps der Weave GitOps innerhalb eines EKS-Clusters verwendet. Ausführliche Informationen finden Sie im GitOpsWeave-Repository.



Wobei:

- Schritt 1: Zusammenführung von Pull-Requests (PR). Ein Entwickler überträgt Änderungen an Kubernetes-Manifesten oder Helm-Diagrammen, die in einem Git-Repository gespeichert sind. Wenn der PR überprüft und mit dem Hauptzweig zusammengeführt wurde, wird der gewünschte Status der Anwendung in der Quellcodeverwaltung aktualisiert.
- Schritt 2: Repository-Synchronisierung. Weave GitOps läuft im Flux-Namespace im EKS-Cluster und überwacht kontinuierlich das konfigurierte Git-Repository. Wenn es Änderungen erkennt, ruft es die neuesten Updates ab, um den deklarierten Status abzugleichen.
- Schritt 3: Bereitstellung im Ziel-Namespace. Weave GitOps vergleicht den gewünschten Status von Git mit dem Live-Status im Cluster. Anschließend werden die erforderlichen Änderungen auf den Ziel-Workload-Namespace angewendet, sodass die Anwendung entsprechend bereitgestellt oder aktualisiert wird.

Jenkins X

Jenkins X ist eine Cloud-native CI/CD Open-Source-Plattform, die Prinzipien für Kubernetes-Umgebungen implementiert. GitOps Jenkins X ist zwar nicht ausschließlich ein GitOps Tool wie Argo CD oder Flux, integriert aber Praktiken in seine Workflows. GitOps

GitOps Unterstützung

Flächen	Funktionen des Tools
GIT-zentrierter Arbeitsablauf	Jenkins X verwendet Git-Repositorys als primäre Informationsquelle sowohl für den Anwendungscode als auch für die Konfigura tion. Alle Änderungen an Anwendungen und Infrastruktur werden über Git vorgenommen.
Umgebung als Code (EaC)	Umgebungen (wie Staging und Produktion) werden in Git-Repositorys als Code definiert . Dies ermöglicht die Versionskontrolle und Überprüfung von Umgebungskonfigurationen.
Automatisierte CI/CD Pipelines	Jenkins X richtet automatisch CI/CD Pipelines für Projekte ein. Diese Pipelines sind als Code (Pipeline als Code) definiert und in Git gespeichert.
Kubernetes-nativ	Jenkins X wurde speziell für Kubernetes-Umgebungen entwickelt. Es verwendet Kubernetes-Ressourcen und benutzerdefinierte Ressourcendefinitionen (). CRDs
Umgebungen in der Vorschau anzeigen	Jenkins X erstellt automatisch temporäre Umgebungen für Pull-Requests. Es ermöglich t die einfache Überprüfung und das Testen von Änderungen vor der Zusammenführung.
Förderung zwischen Umgebungen	Jenkins X verwendet einen GitOps Ansatz zur Förderung von Anwendungen zwischen Umgebungen (z. B. vom Staging bis zur Produktion). Werbeaktionen werden mithilfe von Pull-Requests abgewickelt, um einen ordnungsgemäßen Überprüfungs- und Genehmigungsprozess sicherzustellen.

Flächen	Funktionen des Tools
Verwaltung von Helm-Diagrammen	Jenkins X verwendet Helm-Diagramme, um Anwendungen zu verpacken und bereitzus tellen. Diagramme werden in Git-Repositorys versionsgesteuert.
Automatisierte Versionierung	Jenkins X verwaltet automatisch die Versionie rung von Anwendungen und Releases. Es verwendet semantische Versionierung und generiert Versionshinweise.
ChatOps Integration	Jenkins X unterstützt gängige ChatOps Operationen. Dies entspricht den GitOps Prinzipien der Automatisierung und Zusammenarbeit.
Erweiterbarkeit	Dieses Tool bietet ein Plugin-System zur Erweiterung der Funktionalität. Es ermöglicht die Integration mit verschiedenen Cloud-nativen Tools.
Infrastructure as Code (IaC)	Jenkins X unterstützt Terraform, CloudForm ation AWS Cloud Development Kit (AWS CDK), und andere IaC-Tools zur Definition und Verwaltung der Infrastruktur. Infrastrukturdefin itionen werden zusammen mit dem Anwendung scode versionsgesteuert.
Automatisierte Rollbacks	Jenkins X unterstützt automatisierte Rollbacks , wenn nach der Bereitstellung Probleme festgestellt werden.
Verwaltung von Secrets	Das Tool lässt sich in externe Secrets-M anagement-Lösungen integrieren, um vertrauli che Informationen sicher zu handhaben.

Flächen	Funktionen des Tools
Beobachtbarkeit	Jenkins X bietet die Integration mit Überwachungs- und Protokollierungstools für die Beobachtbarkeit.
Multi-Cloud-Unterstützung	Jenkins X ist so konzipiert, dass es mit verschiedenen Cloud-Anbietern und lokalen Umgebungen funktioniert.
Zusammenarbeit im Team	Dieses Tool fördert die Zusammenarbeit durch Git-basierte Workflows und Pull-Requests.
Kontinuierliches Feedback	Das Tool bietet schnelles Feedback zu Änderungen durch automatisierte Test- und Vorschauumgebungen.
DevOps bewährte Verfahren	Jenkins X implementiert standardmäßig DevOps bewährte Verfahren, einschließlich GitOps Prinzipien.
Deklarative Konfiguration	Das Tool verwendet deklarative Konfigurationen zur Definition von Anwendungen und Umgebungen.
Automatisierte Upgrades	Jenkins X bietet Tools zur Automatisierung von Upgrades der Jenkins X-Plattform selbst.

Jenkins X implementiert diese GitOps Prinzipien, um eine umfassende CI/CD-Lösung für Kubernetes zu erstellen. Es zielt darauf ab, den gesamten Softwarebereitstellungsprozess, vom Code-Commit bis zur Produktionsbereitstellung, zu automatisieren und zu rationalisieren und dabei die üblichen Verfahren einzuhalten. GitOps Auf diese Weise unterstützt es Teams dabei, schnellere, zuverlässigere und konsistentere Implementierungen in cloudnativen Umgebungen zu erreichen.

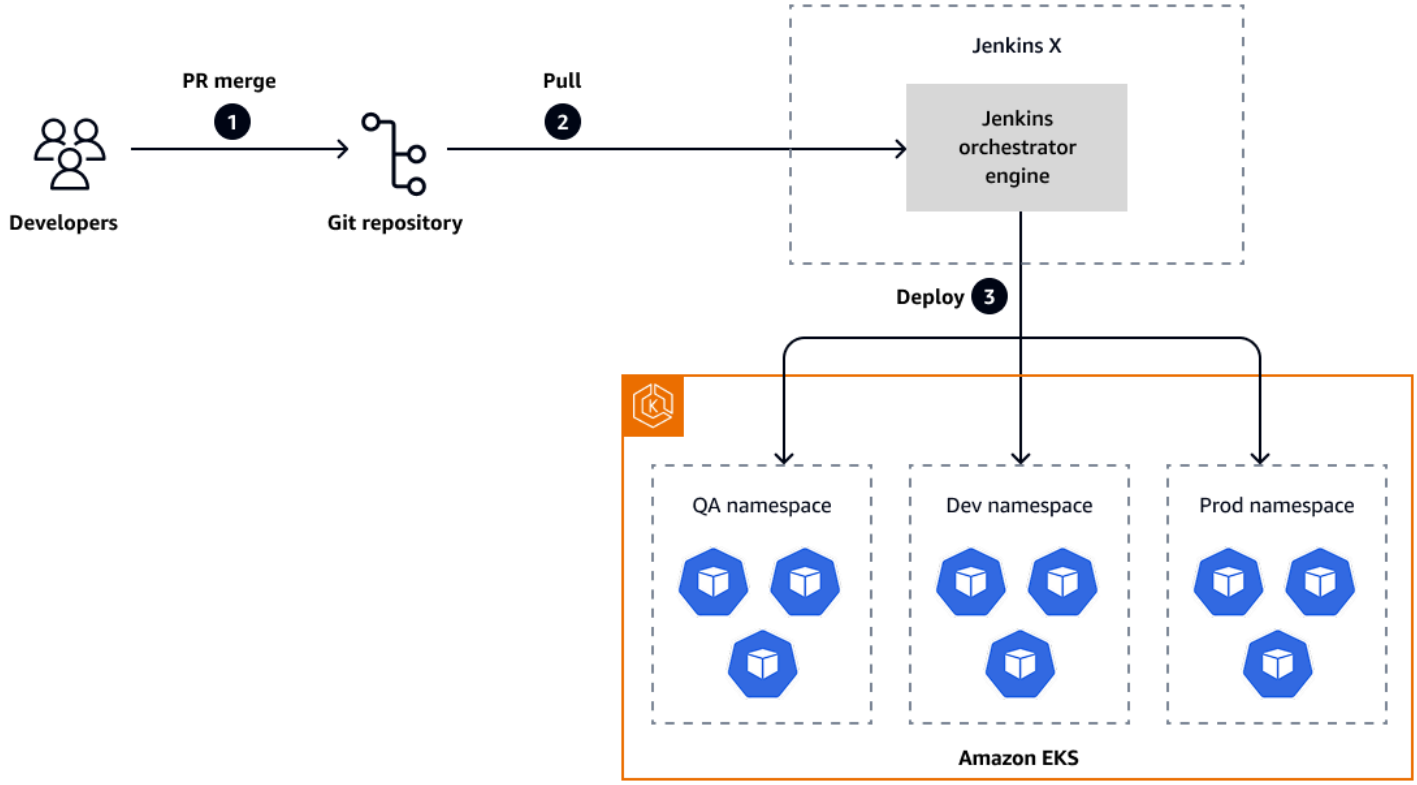
Der Hauptunterschied zwischen Jenkins X und Tools wie Argo CD oder Flux besteht darin, dass Jenkins X eine umfassendere CI/CD Lösung bietet, einschließlich Build-Automatisierung und Pipeline-Management, während gleichzeitig Prinzipien für die Bereitstellung und das Umgebungsmanagement berücksichtigt GitOps werden. Dies macht es besonders für Teams

geeignet, die eine all-in-one Lösung benötigen, die sowohl CI- als auch CD-Aspekte in einem einzigen Framework abdeckt. GitOps

Weitere Informationen finden Sie in der Jenkins X-Dokumentation.

Architektur

Das folgende Diagramm veranschaulicht einen GitOps CD-Workflow, der Jenkins X verwendet. Ausführliche Informationen finden Sie in der Jenkins X-Dokumentation.



Wobei:

- Schritt 1: Zusammenführung von Pull-Requests (PR). Ein Entwickler erstellt eine Pull-Anfrage, die Änderungen an Kubernetes-Manifesten, Helm-Diagrammen oder Anwendungscode enthält, der in einem Git-Repository gespeichert ist. Nach der Überprüfung und Genehmigung wird der PR mit dem Hauptzweig zusammengeführt und aktualisiert den gewünschten Status in der Quellcodeverwaltung.
- Schritt 2: Repository-Synchronisierung. Jenkins X löst automatisch eine CI/CD Pipeline aus, wenn es die Änderung erkennt. Die Pipeline erstellt, testet und verbreitet die Anwendung mithilfe von Prinzipien in verschiedenen Umgebungen (z. B. Staging und Produktion). GitOps

- Schritt 3: Bereitstellung in Ziel-Namespaces. Jenkins X aktualisiert die Umgebungs-Repositorys (Staging und Produktion) mit den neuen Anwendungsversionen. Der Cluster gleicht die Änderungen automatisch ab, indem er die neuesten Manifeste aus Git abruft und die Anwendung in den entsprechenden Namespaces bereitstellt.

GitLab CI/CD

GitLab CI/CD is an integrated part of the GitLab platform that provides continuous integration, delivery, and deployment capabilities. Although GitLab CI/CDist nicht ausschließlich ein GitOps Tool. Sie können es so konfigurieren, dass es GitOps Prinzipien implementiert, insbesondere wenn Sie es für Kubernetes-Bereitstellungen verwenden.

GitOps Unterstützung

Flächen	Funktionen des Tools
Git als zentrale Informationsquelle	GitLab CI/CD verwendet Git-Repositorys, um sowohl Anwendungscode als auch Infrastrukturkonfigurationen zu speichern. Alle Änderungen am System werden über Git vorgenommen, wodurch ein vollständiger Verlauf und ein Audit-Trail gewährleistet sind.
Deklarative Konfiguration	GitLab CI/CD-Pipelines werden in einer .gitlab-ci.yml-Datei definiert, einer deklarativen Konfiguration, die im Git-Repository gespeicherte ist. Kubernetes-Manifeste, Helm-Charts oder andere Infrastructure-as-Code-Dateien (IaC) können im selben Repository gespeicherte werden, um den gewünschten Status der Infrastruktur zu definieren.
Automatisierte Pipelines	GitLab CI/CD löst automatisch Pipelines aus, wenn Änderungen in das Repository übertragen werden. Diese Pipelines können Phasen für die Erstellung, das Testen und die Bereitstellung von Anwendungen umfassen.

Flächen	Funktionen des Tools
Kubernetes-Integration	GitLab CI/CD bietet eine native Kubernetes-Integration und unterstützt GitOps Bereitstellungen im Stil von Kubernetes-Clustern. Es kann automatisch Kubernetes-Ressourcen basierend auf der Konfiguration in Git erstellen und verwalten.
Verwaltung der Umgebung	GitLab CI/CD unterstützt die Definition mehrerer Umgebungen (wie Staging und Produktion) als Code. Bereitstellungen in diesen Umgebungen können automatisiert werden oder erfordern möglicherweise eine manuelle Genehmigung, sofern die geltenden Verfahren eingehalten werden. GitOps
Bewerbungen überprüfen	GitLab kann automatisch temporäre Umgebungen für Merge-Anfragen erstellen, ähnlich wie Vorschauumgebungen in anderen GitOps Tools. Dies ermöglicht das einfache Überprüfen und Testen von Änderungen vor Zusammenführungen.
Fortlaufende Bereitstellung	GitLab CI/CD kann so konfiguriert werden, dass Änderungen automatisch an Kubernetes-Clustern bereitgestellt werden, wenn Änderungen in bestimmten Branches zusammengeführt werden.
IaC	GitLab CI/CD unterstützt die Integration mit Tools wie Terraform und die Verwaltung der Infrastruktur als Code. CloudFormation Infrastrukturdefinitionen können zusammen mit dem Anwendungscode versionsgesteuert werden.

Flächen	Funktionen des Tools
Beobachtbarkeit und Überwachung	GitLab CI/CD bietet integrierte Überwachungs- und Beobachtbarkeitsfunktionen, einschließlich der Integration mit Prometheus und Grafana.
Sicherheitsscans	GitLab CI/CD includes built-in security scanning tools that can be integrated into the CI/CDPipe line zur Durchsetzung der Sicherheit als Teil des GitOps Workflows.
Container-Registrierung	GitLab CI/CD enthält eine integrierte Container -Registry für die nahtlose Integration der Container-Imageverwaltung in den GitOps Workflow.
Automatisch DevOps	Die automatische DevOps Funktion in GitLab CI/CD can automatically configure CI/CD Pipelines, die den GitOps Prinzipien für Kubernetes-Bereitstellungen folgen.
Workflows für Genehmigungen	GitLab CI/CD unterstützt Genehmigungsverfah ren für Bereitstellungen, die eine kontrollierte Beförderung zwischen Umgebungen ermöglich en.
Verwaltung von Secrets	GitLab CI/CD provides features to securely manage and use secrets within CI/CDPipelines.
Versionierung und Veröffentlichungen	GitLab CI/CD supports automatic versioning and release management as part of the CI/ CDProzess.
Rollbacks	GitLab CI/CD ermöglicht einfache Rollbacks zu früheren Versionen, falls nach der Bereitste llung Probleme festgestellt werden.

Flächen	Funktionen des Tools
Auditprotokolle	GitLab CI/CD bietet umfassende Auditprot okolle für alle Aktionen, um den Aspekt der Rückverfolgbarkeit von zu unterstützen. GitOps
Pipelines für mehrere Projekte	GitLab CI/CD unterstützt komplexe GitOps Workflows, die sich über mehrere Projekte oder Repositorien erstrecken.
ChatOps	GitLab CI/CD unterstützt ChatOps Integrati onen, die Zusammenarbeit und Abläufe über Chat-Schnittstellen ermöglichen.
Verwaltung von Kubernetes-Clustern	GitLab CI/CD bietet Funktionen für die Verwaltung von Kubernetes-Clustern direkt über die Schnittstelle. GitLab

Das Kubernetes-Management macht es jedoch GitLab CI/CD is not exclusively designed for GitOps, it can be used effectively to implement GitOps practices, especially for teams that already use GitLab as their primary development platform. Its integrated approach, which combines source control, CI/CD zu einem leistungsstarken Tool für die Implementierung von Workflows. GitOps

Der Hauptunterschied zwischen GitLab CI/CD and dedicated GitOps tools such as Argo CD or Flux is that GitLab provides a more comprehensive platform that includes source control management, issue tracking, and other development tools along with its CI/CD den Funktionen. Dies macht es besonders für Teams geeignet, die eine all-in-one Lösung benötigen, mit der GitOps Praktiken in einem breiteren Entwicklungssystem implementiert werden können.

Weitere Informationen zu GitLab CI/CD und seiner Architektur finden Sie in der GitLab CI/CD-Dokumentation.

Spinnaker

Spinnaker ist zwar nicht ausschließlich als GitOps Tool konzipiert, aber Sie können es so konfigurieren, dass es GitOps Prinzipien implementiert, insbesondere wenn Sie es für Cloud-native und Kubernetes-Bereitstellungen verwenden.

GitOps Unterstützung

Flächen	Funktionen des Tools
Deklarative Konfiguration	Spinnaker verwendet deklarative Pipeline-Definitionen, die normalerweise als JSON- oder YAML-Dateien gespeichert werden. Diese Pipeline-Definitionen können in Git-Repositorys versionsgesteuert werden, was den Gepflogen heiten entspricht. GitOps
IaC	Spinnaker unterstützt die Definition von Infrastruktur- und Bereitstellungskonfiguratio nen als Code. Diese Definitionen können in Git-Repositorys gespeichert werden und können als zentrale Informationsquelle dienen.
Multi-Cloud-Bereitstellungen	Spinnaker ist so konzipiert, dass es mit mehreren Cloud-Anbietern und Kubernetes-Clustern funktioniert. Es ermöglicht konsistente GitOps Praktiken in verschiedenen Umgebunge n.
Pipeline als Code	Spinnaker-Pipelines können als Code definiert und in Git-Repositorys gespeichert werden. Dies ermöglicht die Versionskontrolle und die Überprüfung der Bereitstellungsprozesse.
Automatisierte Bereitstellungen	Sie können Spinnaker so konfigurieren, dass Bereitstellungen auf der Grundlage von Änderungen in Git-Repositorys automatis ch gestartet werden. Das Tool unterstützt kontinuierliche Bereitstellungspraktiken, die für von zentraler Bedeutung sind. GitOps
Unveränderliche Infrastruktur	Spinnaker fördert die Verwendung einer unveränderlichen Infrastruktur, was ein Schlüsselkonzept von ist. GitOps Es fördert

Flächen	Funktionen des Tools
	die Bereitstellung neuer Instanzen, anstatt bestehende Instanzen zu modifizieren.
Rollbacks und Versionierung	Spinnaker bietet robuste Rollback-Funktione n und ermöglicht eine schnelle Rückkehr zu früheren, als funktionierend bekannten Status. Es unterstützt die Versionierung von Bereitste llungen gemäß den Grundsätzen der Rückverfo lgbarkeit. GitOps
Workflows für Genehmigungen	Spinnaker integriert manuelle Beurteilu ngsphasen in Pipelines, um kontrollierte Werbeaktionen zwischen Umgebungen zu unterstützen. Dies unterstützt GitOps Praktiken der Trennung zwischen Bereitstellungen und Releases.
Canary und Bereitstellungen blue/green	Spinnaker unterstützt fortschrittliche Einsatzst rategien, die sich an den GitOps Praktiken für sichere und kontrollierte Freisetzungen orientier en.
Integration mit Versionskontrollsystemen	Spinnaker kann in verschiedene Git-Anbie ter integriert werden, um Pipelines auf der Grundlage von Repository-Ereignissen zu starten.
Kubernetes-Integration	Spinnaker bietet native Unterstützung für Kubernetes und unterstützt GitOps die Verwaltung von Kubernetes-Ressourcen im Stil von Kubernetes.
Verwaltung von Artefakten	Spinnaker unterstützt Artefaktmanagement und Versionierung, die für die Aufrechte rhaltung eines Workflows von entscheidender Bedeutung sind. GitOps

Flächen	Funktionen des Tools
Beobachtbarkeit und Überwachung	Spinnaker bietet die Integration mit Überwachungstools, um den Aspekt der Beobachtbarkeit von zu unterstützen. GitOps
Prüfpfad	Spinnaker bietet detaillierte Bereitstellungsprotokolle und -verläufe, die das Prinzip der Überprüfbarkeit von unterstützen. GitOps
Rollenbasierte Zugriffskontrolle (RBAC)	Dieses Tool implementiert RBAC für eine genaue Kontrolle darüber, wer welche Aktionen ausführen kann, und zwar in Übereinstimmung mit den Sicherheitspraktiken. GitOps
Templating und Parametrisierung	Spinnaker unterstützt Templates in Pipeline-Definitionen, um wiederverwendbare und parametrisierte Bereitstellungen zu ermöglichen.
Förderung der Umwelt	Spinnaker ermöglicht die kontrollierte Förderung von Anwendungen zwischen Umgebungen (z. B. von der Bereitstellung bis zur Produktion).
Integration mit CI-Tools	Spinnaker kann in verschiedene CI-Tools (Continuous Integration) integriert werden, um eine vollständige CI/CD Pipeline bereitzustellen, die den Prinzipien entspricht. GitOps
Benutzerdefinierte Stufen und Erweiterungen	Dieses Tool unterstützt benutzerdefinierte Phasen und Erweiterungen, sodass Teams GitOps Workflows implementieren können, die auf ihre Bedürfnisse zugeschnitten sind.
Zentralisiertes Management	Spinnaker bietet eine zentralisierte Plattform für die Verwaltung von Bereitstellungen in mehreren Umgebungen und Cloud-Anbietern.

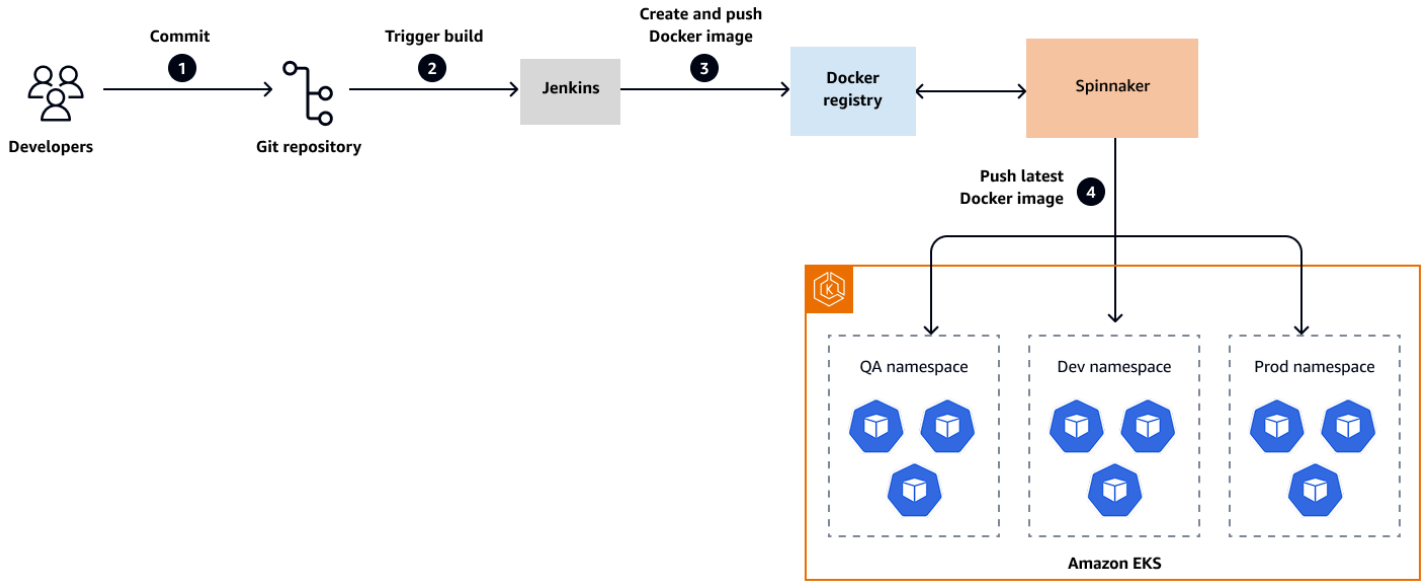
Obwohl Spinnaker nicht in erster Linie als GitOps Tool vermarktet wird, ist es aufgrund seiner Flexibilität und seines robusten Funktionsumfangs in der Lage, GitOps Workflows zu implementieren, insbesondere in komplexen Multi-Cloud-Umgebungen. Der Hauptunterschied zwischen Spinnaker und speziellen GitOps Tools wie Argo CD oder Flux besteht darin, dass Spinnaker eine umfassendere Continuous-Delivery-Plattform mit fortschrittlichen Bereitstellungsstrategien und Multi-Cloud-Unterstützung anbietet.

Die Stärke von Spinnaker liegt in seiner Fähigkeit, komplexe Bereitstellungsszenarien bei verschiedenen Cloud-Anbietern zu bewältigen, und in der Unterstützung fortschrittlicher Bereitstellungsstrategien. Wenn Spinnaker richtig konfiguriert ist, kann es Prinzipien effektiv umsetzen. GitOps Dies macht es zu einem leistungsstarken Tool für Unternehmen, die GitOps Praktiken in unterschiedlichen und komplexen Umgebungen anwenden möchten.

Weitere Informationen finden Sie in der Spinnaker-Dokumentation.

Architektur

Das folgende Diagramm veranschaulicht einen GitOps CD-Workflow, der Spinnaker und Jenkins X verwendet. Ausführliche Informationen finden Sie in der Spinnaker-Dokumentation.



Wobei:

- Schritt 1: Code-Commit. Entwickler übertragen Änderungen am Anwendungscode in ein Git-Repository. Diese Änderungen könnten Aktualisierungen der Anwendung selbst, Dockerfiles oder Kubernetes-Manifeste beinhalten.

- Schritt 2: Jenkins-Build und Image-Erstellung. Jenkins wird automatisch vom Git-Repository über einen Webhook oder Polling ausgelöst. Jenkins erstellt die Anwendung, erstellt ein Docker-Image und überträgt das erstellte Image in eine konfigurierte Docker-Registry (wie Amazon ECR oder Docker Hub).
- Schritt 3: Spinnaker-Bildüberwachung und Pipeline-Trigger. Spinnaker überwacht die Docker-Registry kontinuierlich auf neue Images. Wenn eine neue Image-Version erkannt wird, löst Spinnaker automatisch eine Pipeline aus, um den Bereitstellungsprozess zu starten.
- Schritt 4: Bereitstellung in Ziel-Namespaces. Spinnaker stellt das neue Docker-Image auf Amazon EKS bereit. Basierend auf den Pipeline-Konfigurationen wird das Image in Ziel-Namespaces im Cluster bereitgestellt. Spinnaker stellt sicher, dass die neueste Anwendungsversion bereitgestellt wird und dabei definierte Bereitstellungsstrategien wie Canary-Deployments eingehalten werden. blue/green

Rancher-Flotte

Rancher Fleet ist eine GitOps-at-scale Lösung, die speziell für die Verwaltung mehrerer Kubernetes-Cluster entwickelt wurde. Sie hält sich strikt an die GitOps Prinzipien und konzentriert sich gleichzeitig auf Skalierbarkeit und Multi-Cluster-Management.

GitOps Unterstützung

Flächen	Funktionen des Tools
Git als zentrale Informationsquelle	Fleet verwendet Git-Repositorys als maßgeblic he Quelle für die Definition des gewünschten Status von Anwendungen und Ressourcen in mehreren Clustern. Alle Konfigurationen, einschließlich Kubernetes-Manifeste, Helm-Charts und benutzerdefinierte Ressourcen, werden in Git gespeichert.
Deklarative Konfiguration	Fleet arbeitet mit deklarativen Beschreibungen des gewünschten Zustands für Anwendung en und Ressourcen. Dabei kann es sich um rohe Kubernetes-YAML, Helm-Diagramme,

Flächen	Funktionen des Tools
	Kustomize-Dateien oder Fleet-spezifische benutzerdefinierte Ressourcen handeln.
Automatisierte Synchronisation	Fleet überwacht die Git-Repositorys kontinuie rlich auf Änderungen. Es wendet automatisch Änderungen auf die Zielcluster an, wenn es Unterschiede zwischen dem Git-Status und dem Cluster-Status erkennt.
Verwaltung mehrerer Cluster	Fleet wurde speziell für die Verwaltung von Bereitstellungen in mehreren Kubernetes-Clustern entwickelt. Es kann Tausende von Clustern von einer einzigen Steuerungsebene aus verwalten.
Kubernetes-native Architektur	Fleet besteht aus einer Reihe von benutzerd efinierten Kubernetes-Ressourcen und Controllern. Es verwendet die Erweiteru ngsmechanismen in Kubernetes für den Betrieb. GitOps
Kontinuierlicher Abgleich	Fleet vergleicht ständig den aktuellen Status von Clustern mit dem gewünschten Status, der in Git definiert ist. Es korrigiert automatisch jede Abweichung, die zwischen diesen Zuständen festgestellt wird.
Gruppierung und Targeting von Clustern	Mit Fleet können Sie Cluster gruppieren und Bereitstellungen auf bestimmte Gruppen oder einzelne Cluster ausrichten. Es unterstützt die konsistente Anwendungsbereitstellung in verschiedenen Umgebungen und Clustertypen.

Flächen	Funktionen des Tools
Mehrschichtige Konfigurationen	Fleet unterstützt mehrschichtige Konfigurationen, die Basiskonfigurationen mit umgebungsspezifischen Overlays bieten. Dies entspricht den GitOps Praktiken zur effizienten Verwaltung mehrerer Umgebungen.
Integration steuern	Fleet bietet native Unterstützung für Helm-Diagramme und ermöglicht die einfache Verwaltung komplexer Anwendungen. Es kann Helm-Releases über GitOps Workflows versionieren und verwalten.
Benutzerdefinierte Ressourcendefinitionen (CRDs)	Fleet verwendet benutzerdefinierte Ressourcen wie GitRepo Bundle, um Bereitstellungen zu definieren. Diese CRDs bieten eine Kubernetes-native Methode zur Definition von Workflows. GitOps
Sicherheit und RBAC	Fleet lässt sich zur Zugriffskontrolle in Kubernetes RBAC integrieren. Es unterstützt die sichere Verwaltung sensibler Informationen und Anmeldeinformationen.
Beobachtbarkeit	Fleet bietet Statusinformationen über den Synchronisationsstatus von Clustern und Anwendungen. Es bietet Einblicke in die GitOps Prozesse in der gesamten Clusterflotte.
Skalierbarkeit	Fleet ist so konzipiert, dass es Tausende von Clustern effizient verwalten kann. Es unterstützt umfangreiche GitOps Operationen in Unternehmensumgebungen.

Flächen	Funktionen des Tools
Abhängigkeitsmanagement	Sie können Abhängigkeiten zwischen verschiedenen Ressourcen und Anwendungen definieren. Fleet stellt sicher, dass bei komplexen Bereitstellungen die richtige Reihenfolge der Abläufe eingehalten wird.
Anpassung und Erweiterbarkeit	Fleet unterstützt benutzerdefinierte Skripts und Lifecycle-Hooks für eine erweiterte Anpassung von Bereitstellungen. Es ermöglicht die Integration in bestehende Tools und Workflows.
Offline- und Air-Gap-Support	Fleet kann in Umgebungen mit eingeschränkter oder keiner Internetverbindung eingesetzt werden. Es unterstützt GitOps Arbeitsabläufe in hochsicheren oder regulierten Umgebungen.
Schrittweise Rollouts	Fleet unterstützt stufenweise Rollouts in mehreren Clustern, was kontrollierte und schrittweise Implementierungsstrategien ermöglicht.
Einheitliche Verwaltungsoberfläche	Fleet bietet eine einzige Oberfläche für die Verwaltung von GitOps Workflows in allen Clustern. Es vereinfacht den Betrieb in komplexen Umgebungen mit mehreren Clustern.
Integration mit anderen Rancher-Tools	Fleet lässt sich in andere Rancher-Tools integrieren, um eine umfassende Kubernetes-Managementlösung bereitzustellen.
Prüfpfad und Einhaltung von Vorschriften	Fleet führt einen klaren Prüfpfad für alle Änderungen und Implementierungen. Es hilft Ihnen, Compliance-Anforderungen durch versionskontrollierte, Git-basierte Abläufe zu erfüllen.

Rancher Fleet implementiert diese GitOps Prinzipien mit einem starken Fokus auf Skalierbarkeit und Multi-Cluster-Management. Das Design eignet sich besonders für Unternehmen, die eine große Anzahl von Kubernetes-Clustern in verschiedenen Umgebungen, Rechenzentren oder Cloud-Anbietern verwalten.

Das wichtigste Unterscheidungsmerkmal von Fleet ist die Fähigkeit, das System skalierbar zu handhaben GitOps . Diese Funktion macht es besonders wertvoll für große Unternehmen oder Anbieter von Managed Services, die zahlreiche Cluster verwalten. Tools wie Argo CD oder Flux werden häufig für die Verwaltung einzelner Cluster verwendet, wohingegen Fleet für die Verwaltung GitOps einer großen Clusterflotte konzipiert ist.

Durch die Einhaltung dieser GitOps Prinzipien bietet Rancher Fleet eine Lösung für Unternehmen, die ein konsistentes, skalierbares und automatisiertes Management von Anwendungen und Ressourcen in einer vielfältigen und großen Kubernetes-Umgebung implementieren möchten.

Weitere Informationen finden Sie in der Fleet-Dokumentation.

Architektur

Informationen zur Architektur und zum Arbeitsablauf finden Sie im Fleet-Repository.

Codefresh

Codefresh ist eine moderne CI/CD Plattform, die GitOps Prinzipien unterstützt, insbesondere für Kubernetes-Implementierungen. Codefresh bietet einen umfassenden CI/CD Funktionsumfang, und seine Fähigkeiten sind bemerkenswert. GitOps

GitOps Unterstützung

Flächen	Funktionen des Tools
Git als zentrale Informationsquelle	Codefresh verwendet Git-Repositorys als maßgebliche Quelle für Anwendungscode, Infrastrukturdefinitionen und Pipeline-Konfigurationen. Alle Änderungen am System werden über Git vorgenommen, wodurch ein vollständiger Verlauf und ein Audit-Trail gewährleistet sind.

Flächen	Funktionen des Tools
Deklarative Konfiguration	Codefresh unterstützt deklarative Pipeline-Definitionen mithilfe von YAML-Dateien, die in Git gespeichert sind. Kubernetes-Manifeste, Helm-Diagramme, CloudFormation Vorlagen und andere IaC-Dateien können in denselben Repositorys versionsgesteuert werden.
GitOps Armaturenbrett	Codefresh bietet ein spezielles GitOps Dashboard zur Visualisierung und Verwaltung GitOps von Workflows. Es bietet einen klaren Überblick über den Synchronisationsstatus zwischen Git- und Cluster-Status.
Automatisierte Synchronisation	Codefresh überwacht die Git-Repositorys kontinuierlich auf Änderungen. Es startet automatisch Pipelines, um Änderungen an den Zielumgebungen vorzunehmen, wenn es Unterschiede feststellt.
Kubernetes-Integration	Codefresh bietet eine umfassende Integrati on mit Kubernetes, um Bereitstellungen im Stil mehrerer Cluster zu unterstützen GitOps. Es unterstützt verschiedene Kubernetes-Ressour cen und benutzerdefinierte Ressourcendefiniti onen (). CRDs
Verwaltung der Umgebung	Sie können mehrere Umgebungen (z. B. Entwicklung, Staging und Produktion) als Code definieren und verwalten. Codefresh unterstüt zt die Werbung zwischen Umgebungen mithilfe GitOps von Methoden.

Flächen	Funktionen des Tools
Integration mit Argo CD	Codefresh lässt sich für erweiterte Funktione n in Argo CD integrieren. GitOps Es kombinier t seine CI-Funktionen mit den CD-Stärken von Argo CD, um eine Komplettlösung bereitzus tellen. GitOps
Helm-Unterstützung	Codefresh unterstützt Helm-Charts und ermöglicht so eine einfache Verwaltung komplexer Anwendungen. GitOps Es bietet auch Versionierung und Werbung für Helm-Charts.
Progressive Lieferung	Codefresh unterstützt fortschrittliche Bereitste llungsstrategien wie Canary und blue/green Deployments. Sie können diese Strategien mithilfe GitOps von Workflows implementieren und verwalten.
Rollbacks und Versionierung	Codefresh ermöglicht einfache Rollbacks zu früheren Versionen, wenn nach der Bereitste llung Probleme festgestellt werden. Es behält die Versionierung der Bereitstellung zur Rückverfolgbarkeit bei.
Workflows für Genehmigungen	Codefresh unterstützt manuelle und automatis ierte Genehmigungsprozesse für Bereitste llungen. Es ermöglicht kontrollierte Werbeakti onen zwischen verschiedenen Umgebunge n, die den geltenden Praktiken entsprechen GitOps.
IaC	Codefresh unterstützt die Integration mit IaC-Tools wie Terraform CloudFormation . Es ermöglicht die Versionskontrolle von Infrastrukturdefinitionen zusammen mit dem Anwendungscode.

Flächen	Funktionen des Tools
Beobachtbarkeit und Überwachung	Codefresh bietet integrierte Überwachungs- und Beobachtbarkeitsfunktionen. Es bietet auch Integrationen mit externen Überwachungstools für eine verbesserte Sichtbarkeit.
Sicherheitsscannen	Codefresh bietet Funktionen für Sicherhei tsscans, die in GitOps Arbeitsabläufe integrier t werden können. Sicherheitsüberprüfungen sind Teil des automatisierten Bereitstellungspro zesses.
Audit-Trails	Codefresh führt umfassende Auditprotokolle für alle Aktionen und Änderungen. Es unterstüt zt die Rückverfolgbarkeits- und Compliance-Aspekte von. GitOps
RBAC und Zugriffskontrolle	Codefresh implementiert eine rollenbasierte Zugriffskontrolle (RBAC) für eine differenz ierte Rechteverwaltung. Dies trägt dazu bei, sichere Abläufe in Teams und Umgebungen zu gewährleisten. GitOps
GitOps -Automatisierung	Codefresh bietet Funktionen zur Automatis ierung verschiedener Aspekte von GitOps Workflows, einschließlich der Erstellung und Zusammenführung von Pull-Requests (PR).
Multi-Cloud- und Hybrid-Bereitstellungen	Codefresh unterstützt GitOps Workflows über mehrere Cloud-Anbieter und lokale Umgebungen hinweg.
Templating und Parametrisierung	Codefresh unterstützt Vorlagen in Pipeline- und Bereitstellungskonfigurationen. Dies ermöglich t wiederverwendbare und parametrisierte Workflows GitOps.

Flächen	Funktionen des Tools
Integriertes Bildmanagement	Codefresh bietet integrierte Funktionen zur Verwaltung von Container-Images. Es integrier t Image-Builds und Bereitstellungen in GitOps Workflows.
GitOps für die Verwaltung von Geheimnissen	Codefresh bietet sichere Möglichkeiten zur Verwaltung von Geheimnissen innerhalb von GitOps Workflows. Es lässt sich in externe Lösungen zur Verwaltung von Geheimnissen integrieren.
Funktionen für die Zusammenarbeit	Codefresh bietet Funktionen für die Teamzusammenarbeit innerhalb von GitOps Prozessen. Zu diesen Funktionen gehören Kommentare, Benachrichtigungen und gemeinsame Dashboards.

Der Codefresh-Ansatz für GitOps zeichnet sich durch die Integration von CI/CD-Funktionen in Praktiken aus. GitOps Es zielt darauf ab, eine umfassende Plattform bereitzustellen, die den gesamten Lebenszyklus der Softwarebereitstellung abdeckt und gleichzeitig die Prinzipien einhält. GitOps

Das Hauptunterscheidungsmerkmal von Codefresh in diesem GitOps Bereich ist sein einheitlicher Plattformansatz, der CI-Funktionen mit CD und Funktionen kombiniert. GitOps Dies macht es besonders für Teams geeignet, die eine all-in-one Lösung suchen, die komplexe CI/CD Szenarien bewältigen und gleichzeitig Praktiken implementieren GitOps kann.

Codefresh bietet eine Plattform für Unternehmen, die GitOps Methoden in einem breiteren CI/CD Kontext anwenden möchten, insbesondere bei der Arbeit mit Kubernetes und Cloud-nativen Technologien.

Weitere Informationen finden Sie in der Codefresh-Dokumentation.

Pulumi

Pulumi ist eine IaC-Plattform, die nicht ausschließlich dafür konzipiert ist. GitOps Sie kann jedoch effektiv zur Implementierung von GitOps Prinzipien eingesetzt werden, insbesondere für Cloud-Infrastruktur- und Kubernetes-Implementierungen.

GitOps Unterstützung

Flächen	Funktionen des Tools
IaC	Mit Pulumi können Sie Ihre Infrastruktur mithilfe von Allzweck-Programmiersprachen wie Python und Go definieren. TypeScript Dieser codebasierte Ansatz entspricht der GitOps Betonung versionierter, deklarativer Konfigura tionen.
Git als zentrale Informationsquelle	Infrastrukturcode in Pulumi kann in Git-Repos itorys gespeichert und versionsgesteuert werden. Dadurch wird sichergestellt, dass Git als zentrale Informationsquelle für Infrastru kturdefinitionen dient.
Deklarativer gewünschter Zustand	Pulumi verwendet zwar Programmiersprache n, beschreibt aber dennoch den gewünscht en Zustand der Infrastruktur deklarativ. Der Code definiert, wie die Infrastruktur aussehen soll, nicht der step-by-step Prozess, um sie zu erstellen.
Automatisierte Synchronisation	Pulumi kann in CI/CD Pipelines integrier t werden, um Änderungen automatisch anzuwenden, wenn Code in Git aktualisi ert wird. Dies ermöglicht die kontinuierliche Implementierung von Infrastrukturänderungen, was ein GitOps Schlüsselprinzip ist.

Flächen	Funktionen des Tools
Multi-Cloud- und Kubernetes-Unterstützung	Pulumi unterstützt eine Vielzahl von Cloud-Anbietern und Kubernetes, sodass Sie den Praktiken in verschiedenen Umgebungen folgen können. GitOps Das Tool ermöglicht eine konsistente Verwaltung von Ressourcen auf verschiedenen Plattformen.
Statusverwaltung	Pulumi verwaltet den Zustand der Infrastru ktur, die remote und sicher gespeichert werden kann. Dieses Zustandsmanagement ist für die GitOps Praxis von entscheidender Bedeutung und gewährleistet die Konsistenz zwischen dem definierten Zustand und dem tatsächlichen Zustand Ihrer Infrastruktur.
Erkennung und Abgleich von Abweichungen	Pulumi kann Unterschiede zwischen dem gewünschten Zustand (im Code) und dem tatsächlichen Zustand der Infrastruktur erkennen. Sie gleicht diese Unterschiede im Einklang mit dem GitOps Grundsatz der kontinuierlichen Aussöhnung aus.
Politik als Code	Mit Pulumi CrossGuard können Sie Richtlini en als Code definieren und durchsetzen. Dies ermöglicht eine versionskontrollierte Verwaltun g von Compliance GitOps - und Sicherhei tsrichtlinien im Stil einer Version.
Verwaltung von Secrets	Pulumi bietet sichere Methoden zur Verwaltung sensibler Informationen innerhalb des Infrastru kturcodes. Es unterstützt die Integration mit externen Geheimnisverwaltungssystemen, was für GitOps Sicherheitspraktiken von entscheid ender Bedeutung ist.

Flächen	Funktionen des Tools
Modulare und wiederverwendbare Komponenten	Pulumi unterstützt die Erstellung wiederverwendbarer Komponenten und Module. Diese Modularität entspricht den GitOps Praktiken zur Verwaltung komplexer Bereitstellungen in mehreren Umgebungen.
Vorschau anzeigen und planen	Pulumi bietet die Möglichkeit, eine Vorschau von Änderungen anzuzeigen, bevor sie übernommen werden. Dies unterstützt den GitOps Grundsatz sicherer, vorhersehbarer Änderungen an der Infrastruktur.
Rollbacks und Geschichte	Pulumi führt eine Historie der Bereitstellungen und unterstützt Rollbacks zu früheren Status. Dies entspricht den GitOps Grundsätzen der Rückverfolgbarkeit und Reversibilität.
Kontinuierliche Bereitstellung für die Infrastruktur	Pulumi kann in CI/CD Pipelines integriert werden, um die kontinuierliche Umsetzung von Infrastrukturänderungen zu gewährleisten. Es unterstützt automatisiertes Testen und Validieren von Infrastrukturcode.
RBAC und Zugriffskontrolle	Pulumi bietet eine rollenbasierte Zugriffskontrolle für die Verwaltung, wer Änderungen an der Infrastruktur vornehmen kann. Dies unterstützt GitOps Sicherheits- und Governance-Praktiken.
Beobachtbarkeit und Protokollierung	Pulumi bietet Funktionen zur Protokollierung und Überwachung von Infrastrukturänderungen. Diese Funktionen unterstützen den Aspekt der Beobachtbarkeit von Praktiken. GitOps

Flächen	Funktionen des Tools
Integration mit anderen Tools	Pulumi kann in verschiedene Tools in der Cloud integriert werden. Diese Flexibilität ermöglicht umfassende GitOps Workflows.
Umweltmanagement	Pulumi unterstützt die Verwaltung mehrerer Umgebungen (Entwicklung, Staging, Produktio n), indem dieselbe Codebasis mit unterschi edlichen Konfigurationen verwendet wird. Dies entspricht den GitOps Praktiken für ein konsistentes Management mehrerer Umgebungen.
Abhängigkeitsmanagement	Pulumi kümmert sich um Abhängigk eiten zwischen Ressourcen und stellt die korrekte Reihenfolge der Abläufe sicher. Dies ist entscheidend für komplexe GitOps Bereitstellungen, die voneinander abhängige Komponenten beinhalten.
Anbieter benutzerdefinierter Ressourcen	Mit Pulumi können Sie benutzerdefinierte Anbieter erstellen, um jeden API-geste uerten Dienst zu verwalten. Dadurch werden die GitOps Praktiken auf eine Vielzahl von Ressourcen ausgedehnt, die über die Standard-Cloud-Angebote hinausgehen.
Funktionen für die Zusammenarbeit	Pulumi unterstützt die Zusammenarbeit im Team durch gemeinsame Status- und Zugriffsk ontrollen. Dies erleichtert GitOps Arbeitsabläufe in Teamumgebungen.

Mithilfe dieser Pulumi-Funktionen können Unternehmen GitOps Verfahren für ihre Infrastruktur implementieren, insbesondere in Szenarien, in denen sie eine detaillierte Steuerung oder komplexe Logik benötigen oder eine Vielzahl von Cloud- und lokalen Ressourcen innerhalb eines einzigen, konsistenten Frameworks verwalten möchten.

Pulumis Ansatz GitOps ist einzigartig, da er die Leistungsfähigkeit und Flexibilität von Allzweck-Programmiersprachen für das Infrastrukturmanagement nutzt und gleichzeitig die Prinzipien einhält. GitOps Dies kann besonders für Teams von Vorteil sein, die es vorziehen, mit vertrauten Programmiersprachen zu arbeiten und bewährte Methoden der Softwareentwicklung auf das Infrastrukturmanagement anwenden möchten.

Das Hauptunterscheidungsmerkmal von Pulumi in GitOps ist die Verwendung von Standard-Programmiersprachen zur Definition der Infrastruktur. Herkömmliche GitOps Tools verwenden häufig YAML oder domänenspezifische Sprachen, wohingegen Pulumi eine komplexere Logik, eine bessere Wiederverwendung von Code und eine einfachere Integration in bestehende Entwicklungsworkflows ermöglicht.

Weitere Informationen finden Sie in der Pulumi-Dokumentation.

GitOps Vergleich der Tools

Hier finden Sie einen Vergleich der neun GitOps Tools, die in den vorherigen Abschnitten besprochen wurden. Berücksichtigen Sie bei der Auswahl eines Tools Ihre spezifischen Anforderungen, die vorhandene Infrastruktur, das Fachwissen des Teams und das gewünschte Maß an Kontrolle und Anpassung.

Benutzerfreundlichkeit

- Argo CD, Flux und Rancher Fleet sind im Allgemeinen einfacher einzurichten.
- Spinnaker und Jenkins X haben steilere Lernkurven.
- Weave erfordert GitOps möglicherweise mehr Einstellungen für erweiterte Funktionen.
- GitLab CI/CD und Codefresh bieten integrierte Erlebnisse.

Kubernetes-Integration

- Argo CD, Flux und Rancher Fleet sind sehr auf Kubernetes ausgerichtet.
- Jenkins X und Weave bieten breitere Funktionen. GitOps DevOps
- Die anderen Tools unterstützen Kubernetes, ohne sich ausschließlich darauf zu konzentrieren.

CI/CD-Funktionen

- Jenkins X, Lösungen. GitLab CI/CD, and Codefresh offer complete CI/CD
- Argo CD, Flux und Weave GitOps konzentrieren sich mehr auf den CD-Aspekt des Workflows und erfordern häufig die Integration mit separaten CI-Tools.

GitOps Reinheit

- Argo CD und Flux sind Tools, die sich speziell auf GitOps Folgendes konzentrieren:
- Die anderen Tools beinhalten GitOps Prinzipien in unterschiedlichem Maße.

Multi-Cloud-Unterstützung

- Spinnaker und Pulumi zeichnen sich in Multi-Cloud-Szenarien aus.
- Die anderen Tools können cloudübergreifend eingesetzt werden, erfordern jedoch möglicherweise eine zusätzliche Einrichtung.

Unterstützung mehrerer Cluster

- Alle Tools unterstützen Bereitstellungen mit mehreren Clustern.
- Argo CD und Weave GitOps verfügen über erweiterte Multi-Cluster-Managementfunktionen.

Integration

- Flux wird stark von der Cloud Native Computing Foundation (CNCF) unterstützt.
- Argo CD hat eine große und aktive Community.
- Argo CD und Flux verfügen über eine starke Kubernetes-Integration.
- Jenkins X verwendet das umfassendere Jenkins-System.
- Weave GitOps ist neuer, wächst aber mit starker kommerzieller Unterstützung.
- GitLab CI/CD lässt sich eng in integrieren. GitLab
- Rancher Fleet funktioniert gut innerhalb des Rancher-Systems.

Gemeinschaft und Unterstützung

- Flux hat eine starke CNCF-Unterstützung.
- Argo CD GitLab, und Spinnaker haben große Gemeinschaften.
- Kommerzieller Support ist für die meisten Tools verfügbar.

Funktionen für Unternehmen

- Weave GitOps und Jenkins X bieten standardmäßig mehr unternehmensorientierte Funktionen.
- Argo CD und Flux sind für Unternehmen erhältlich oder können für den Einsatz in Unternehmen erweitert werden.

Flexibilität und Erweiterbarkeit

- Flux ist hochmodular und erweiterbar.
- Argo CD bietet gute Anpassungsmöglichkeiten.
- Jenkins X ist sehr erweiterbar, erfordert jedoch möglicherweise mehr Aufwand.
- Weave GitOps zielt darauf ab, eine Komplettlösung mit weniger Erweiterbarkeit anzubieten.

Skalierbarkeit

- Spinnaker und GitLab CI/CD sind bekannt für ihre Skalierbarkeit auf Unternehmensebene.
- Argo CD und Flux eignen sich gut für groß angelegte Kubernetes-Implementierungen.

Verwaltung der Infrastruktur

- Pulumi konzentriert sich auf das Infrastrukturmanagement.
- Weave GitOps und Flux bieten gute IaC-Funktionen.

Programmiermodell und Sprachunterstützung

- In Pulumi können Sie die Infrastruktur mithilfe von Allzweck-Programmiersprachen wie Python, Go TypeScript, C# und Java definieren. Pulumis Verwendung von Standardsprachen ermöglicht die Integration von Infrastrukturcode in vertraute Entwicklungsabläufe, Testpraktiken und komplexe Logik.
- Terraform verwendet die HashiCorp Konfigurationssprache (HCL).
- CloudFormation verwendet JSON- und YAML-Vorlagen.
- Argo CD, Flux, Rancher Fleet, Weave GitOps, Spinnaker und GitLab CI/CD verwalten hauptsächlich YAML- oder deklarative Konfigurationsdateien.
- Jenkins X verwaltet YAML- und skriptbasierte Pipelines, bietet aber keine systemeigene Allzweckprogrammierung für IaC.

Anwendungsfälle für Argo CD und Flux

Dieser Abschnitt konzentriert sich auf zwei Tools, Argo CD und Flux, die reine GitOps Funktionalität bieten. In diesem Zusammenhang GitOps bezieht sich Pure auf ein Modell, bei dem ein Git-Repository als zentrale Informationsquelle für den gewünschten Status von Anwendungen und Infrastruktur dient. Alle Änderungen werden über Git-Commits vorgenommen, und das System synchronisiert die Live-Umgebung automatisch so, dass sie dem im Repository definierten Status entspricht. Außerhalb von Git-Vorgängen ist kein manuelles Eingreifen erforderlich.

Allgemeine Überlegungen

- Möglicherweise bevorzugen Sie die Verwendung von Argo CD in Umgebungen, in denen visuelles Management und anwendungsorientierte Workflows wichtig sind.
- Sie könnten sich für Flux entscheiden, wenn Sie leichtere Lösungen, eine starke Mehrmandantenfähigkeit oder eine tiefe Integration in das breitere Netzwerk der Cloud Native Computing Foundations (CNCF) benötigen.
- Argo CD spricht aufgrund seiner intuitiven Benutzeroberfläche häufig Teams an, die von herkömmlichem CI/CD auf umsteigen. GitOps
- Flux wird häufig in Cloud-nativen Umgebungen bevorzugt, in denen CLI-basierte Workflows und IaC-Praktiken bereits etabliert sind.

Letztlich hängt die Wahl zwischen Argo CD und Flux oft von Ihren spezifischen organisatorischen Anforderungen, den vorhandenen Tools und den Präferenzen Ihres Teams ab. Beide Tools sind in der Lage, die meisten GitOps Szenarien zu bewältigen. Wir empfehlen Ihnen daher, sie auf der Grundlage Ihrer spezifischen Anwendungsfälle und Anforderungen zu bewerten.

Anwendungsfälle für Argo CD

Visuelles Management:

- Wenn Sie eine benutzerfreundliche Benutzeroberfläche benötigen, um Bereitstellungen zu verwalten und Anwendungsstatus zu visualisieren.
- Für Teams, die eine grafische Oberfläche für die Überwachung und Fehlerbehebung bevorzugen.

Anwendungsorientierter Ansatz:

- Wenn Sie Bereitstellungen auf Anwendungsebene verwalten möchten, anstatt einzelne Ressourcen zu verwalten.
- Für Organisationen, die ihre Bereitstellungen nach Anwendungskonzepten strukturieren.

Verwaltung mehrerer Cluster:

- Wenn die Verwaltung von Bereitstellungen über mehrere Cluster hinweg eine Hauptanforderung ist.
- Für komplexe, verteilte Umgebungen mit vielen Clustern.

Rollback- und Synchronisierungswellen:

- Wenn Sie eine genaue Kontrolle über den Bereitstellungsprozess benötigen, einschließlich Synchronisierungswellen und manueller Eingriffe.
- Für Szenarien, die komplexe Rollback-Strategien erfordern.

Integration mit vorhandenen Tools:

- Wenn Sie im Argo-Projekt bereits andere Tools wie Argo Workflows und Argo Events verwenden.

Unternehmensumgebungen:

- Für große Unternehmen, die standardmäßig eine robuste RBAC- und Single Sign-On-Integration benötigen.

Flux-Anwendungsfälle

Einfache Bereitstellungen:

- Wenn Sie eine leichtere, weniger GitOps ressourcenintensive Lösung benötigen.
- Für Edge-Computing- oder IoT-Szenarien, in denen Ressourcen eingeschränkt sein könnten.

Automatisierte Image-Updates:

- Wenn die automatische Erkennung und Bereitstellung neuer Container-Images eine zentrale Anforderung ist.

- Für Teams, die sich auf die kontinuierliche Bereitstellung mit häufigen Image-Updates konzentrieren.

Mehrmandantenfähigkeit:

- Wenn eine starke Unterstützung für mehrere Mandanten erforderlich ist, insbesondere in gemeinsam genutzten Cluster-Umgebungen.
- Für Dienstanbieter oder große Organisationen, in denen Teams oder Projekte strikt voneinander getrennt sind.

IaC:

- Wenn es wichtig ist, sowohl Anwendungen als auch Infrastruktur über denselben GitOps Workflow zu verwalten.
- Für Teams, die stark in das IaC-Paradigma investiert haben.

Helmintegration:

- Wenn die umfangreiche Verwendung von Helm-Diagrammen Teil Ihrer Bereitstellungsstrategie ist.
- Für Umgebungen mit komplexen Helm-basierten Bereitstellungen.

CNCF-Projektintegration:

- Wenn eine enge Integration mit anderen CNCF-Projekten wichtig ist.
- Für Organisationen, die sich an den Technologien und Prinzipien von CNCF orientieren.

Modulare Architektur:

- Wenn Sie Flexibilität benötigen, um nur bestimmte Komponenten des GitOps Toolkits zu verwenden.
- Für Teams, die mithilfe modularer Komponenten benutzerdefinierte GitOps Workflows erstellen möchten.

Progressive Bereitstellung:

- Wenn fortschrittliche Bereitstellungsstrategien wie Canary-Releases oder A/B Tests zu den Kernanforderungen gehören.

Vergleich der Funktionen

Flächen	Argo CD	Flux
Support der GitOps Kernprinz ipien	 Ja	 Ja
Architektur	End-to-end Anwendung zur Implementierung von Kubernetes-Workflows GitOps	Bietet Kubernetes CRDs und Controller für GitOps
Aufstellen	Einfach	Komplex
Unterstützung für Helme	 Ja	 Ja
Passen Sie den Support an	 Ja	 Ja
Integrierte GUI	CLI und Web-UI mit vollem Funktionsumfang	CLI und optionales, leichtes Webinterface
RBAC-Unterstützung	Granulare Steuerung	Natives RBAC von Kubernetes
Multi-Tenancy- und Multi-Clu ster-Unterstützung	Hervorragende Unterstützung für Multi-Cluster	Hervorragende Unterstützung für Mehrmandantenfähigkeit
Authentifizierung mit einmalige r Anmeldung	 Ja	 Ja

Flächen	Argo CD	Flux
Synchronisieren Sie die Automatisierung	Fähigkeit, Fenster zu synchronisieren	Möglichkeit, Abstimmungsintervalle festzulegen
Teilweise Synchronisierung	 Ja	 Nein
Prozess der Versöhnung	Unterstützt manuelle und automatische Synchronisierungen. Es stehen verschiedene Strategien zur Verfügung.	Unterstützt manuelle und automatische Synchronisierungen.
Erweiterbarkeit	Unterstützt benutzerdefinierte Plugins. Eingeschränkte Anpassungsmöglichkeiten.	Unterstützt benutzerdefinierte Controller. Gute Erweiterbarkeit und Integrationen von Drittanbietern.
Community-Unterstützung	Große und aktive Community.	Kleinere, aber wachsende Gemeinschaft.
Skalierbarkeit	Gute Skalierbarkeit, aber begrenzt durch die Datenabrufrate der Weboberfläche. Community-Analysen deuten darauf hin, dass Zehntausende von Anwendungen unterstützt werden.	Klare Anleitungen für die horizontale und vertikale Skalierbarkeit von bis zu Zehntausenden von Anwendungen.

Bewährte Methoden für die Auswahl eines GitOps Tools

Dieser Abschnitt enthält Überlegungen, Tipps und bewährte Methoden zur Auswahl eines GitOps Tools für Ihren EKS-Cluster. Die richtige Wahl hängt von Ihrem spezifischen Kontext, Ihren Anforderungen und Ihrer langfristigen Strategie ab. Oft ist es von Vorteil, einen Machbarkeitsnachweis mit Ihren wichtigsten Entscheidungen durchzuführen, bevor Sie eine endgültige Entscheidung treffen.

Beurteilen Sie die Bedürfnisse und Fähigkeiten Ihres Unternehmens:

- Berücksichtigen Sie die aktuellen Fähigkeiten und die Bereitschaft Ihres Teams, neue Tools zu erlernen.
- Bewerten Sie die Komplexität Ihrer Amazon EKS-Umgebung. (Verwenden Sie beispielsweise einen einzelnen Cluster oder mehrere Cluster?)
- Ermitteln Sie Ihre spezifischen Anforderungen an Compliance, Sicherheit und Skalierbarkeit.

> **Bewährte Methode**
>
> Erstellen Sie ein detailliertes Anforderungsdokument, in dem die erforderlichen Funktionen und nützliche, aber nicht erforderliche Funktionen beschrieben werden.

Bewerten Sie den Reifegrad und die Akzeptanz des Tools:

- Untersuchen Sie den Reifegrad potenzieller GitOps Tools und deren Akzeptanz in der Branche.
- Suchen Sie nach Tools, die sich in Amazon EKS-Umgebungen bewährt haben.

> **Bewährte Methode**
>
> Priorisieren Sie Tools, die weit verbreitet sind und eine starke Präsenz im Netzwerk der Cloud Native Computing Foundation (CNCF) haben.

Erwägen Sie die Integration mit Ihrer bestehenden Toolchain:

- Beurteilen Sie, wie gut sich das GitOps Tool in Ihre aktuelle CI/CD Pipeline, Überwachungslösungen und andere betriebliche Tools integrieren lässt.

- Suchen Sie nach systemeigenen Integrationen AWS-Services wie IAM, Amazon ECR und. CloudWatch

> **Bewährte Methode**
>
> Erstellen Sie einen Machbarkeitsnachweis, um die Integrationsfunktionen zu testen, bevor Sie eine endgültige Entscheidung treffen.

Bewerten Sie die Sicherheitsfunktionen:

- Priorisieren Sie Tools, die über robuste Funktionen für die rollenbasierte Zugriffskontrolle (RBAC) verfügen und sich gut in IAM integrieren lassen.
- Suchen Sie nach Funktionen, die die sichere Verwaltung von Geheimnissen und die Durchsetzung von Richtlinien unterstützen.

> **Bewährte Methode**
>
> Wählen Sie ein Tool, das fundierte GitOps Sicherheitspraktiken unterstützt, darunter Policy-as-Code und automatisierte Konformitätsprüfungen.

Beurteilen Sie Skalierbarkeit und Leistung:

- Überlegen Sie, wie das Tool bei einer großen Anzahl von Anwendungen und Clustern abschneidet.
- Bewerten Sie die Auswirkungen auf die Clusterleistung und den Ressourcenverbrauch.

> **Bewährte Methode**
>
> Führen Sie Leistungstests mit Workloads durch, die Ihrer Produktionsumgebung ähneln, um sicherzustellen, dass das Tool Ihrer Größenordnung gewachsen ist.

Erwägen Sie die Unterstützung mehrerer Cluster und mehrerer Umgebungen:

- Wenn Sie über mehrere EKS-Cluster verfügen oder dies planen, sollten Sie Tools priorisieren, die über leistungsstarke Verwaltungsfunktionen für mehrere Cluster verfügen.

- Suchen Sie nach Funktionen, die konsistente Bereitstellungen in verschiedenen Umgebungen (z. B. Entwicklung, Staging und Produktion) unterstützen.

> **Bewährte Methode**
>
> Wählen Sie ein Tool, das die zentrale Verwaltung mehrerer Cluster unter Beibehaltung der umgebungsspezifischen Konfigurationen ermöglicht.

Evaluieren Sie die Beobachtbarkeits- und Überwachungsfunktionen:

- Suchen Sie nach Tools, die einen klaren Überblick über den Status Ihrer Bereitstellungen und den Zustand Ihres Clusters bieten.
- Überlegen Sie, wie gut sich das Tool in Ihre bestehenden Überwachungs- und Protokollierungslösungen integrieren lässt.

> **Bewährte Methode**
>
> Priorisieren Sie Tools, die anpassbare Dashboards und Warnmechanismen für die proaktive Problemerkennung bieten.

Beurteilen Sie die Lernkurve und die Dokumentation:

- Beurteilen Sie die Qualität und Vollständigkeit der Dokumentation des Tools.
- Berücksichtigen Sie die Verfügbarkeit von Schulungsressourcen und die Unterstützung durch die Community.

> **Bewährte Methode**
>
> Wählen Sie ein Tool mit gut gepflegter Dokumentation, aktiven Community-Foren und offiziellen Schulungsprogrammen oder Zertifizierungen.

Berücksichtigen Sie die Kosten und die Ressourcennutzung:

- Bewerten Sie sowohl die direkten Kosten (wie Lizenzierung und Support) als auch die indirekten Kosten (wie Betriebskosten und Schulungskosten), die mit der Einführung des Tools verbunden sind.
- Beurteilen Sie die Effizienz des Tools in Bezug auf den Verbrauch von Rechen- und Speicherressourcen.

> **Bewährte Methode**
>
> Führen Sie eine Analyse der Gesamtbetriebskosten (TCO) durch, die sowohl kurzfristige als auch langfristige Kosten umfasst.

Evaluieren Sie Flexibilität und Anpassungsoptionen:

- Suchen Sie nach Tools, mit denen Sie Workflows an Ihre spezifischen Bedürfnisse anpassen können.
- Berücksichtigen Sie die Erweiterbarkeit des Tools durch Plugins oder APIs.

> **Bewährte Methode**
>
> Wählen Sie ein Tool, das Standardfunktionen mit der Möglichkeit zur Anpassung an Ihre individuellen Anforderungen in Einklang bringt.

Beurteilen Sie die Funktionen für kontinuierliche Bereitstellung und progressive Bereitstellung:

- Suchen Sie nach Tools, die fortschrittliche Bereitstellungsstrategien wie Canary-Versionen und blue/green Bereitstellungen unterstützen.
- Prüfen Sie, wie einfach die Implementierung und Verwaltung dieser Strategien ist.

> **Bewährte Methode**
>
> Priorisieren Sie Tools, die integrierte Unterstützung für progressive Bereitstellungsmuster bieten, um das Risiko bei Ihren Implementierungen zu minimieren.

Ziehen Sie Herstellerbindung und Portabilität in Betracht:

- Beurteilen Sie die Abhängigkeiten des Tools von bestimmten Cloud-Anbietern oder -Technologien.
- Denken Sie darüber nach, wie einfach es ist, in future bei Bedarf zu einem anderen Tool zu migrieren.

> **Bewährte Methode**
>
> Bevorzugen Sie Tools, die offene Standards verwenden und Exportfunktionen für Ihre GitOps Konfigurationen bieten.

Evaluieren Sie den Community-Support und die Erweiterungen:

- Schauen Sie sich die Größe und Aktivität der Benutzergemeinschaft an.
- Beurteilen Sie die Verfügbarkeit von Integrationen und Plugins von Drittanbietern.

> **Bewährte Methode**
>
> Treten Sie Community-Foren oder Benutzergruppen bei, um Erfahrungen aus erster Hand von anderen Benutzern zu erhalten, bevor Sie eine Entscheidung treffen.

Beachten Sie die Compliance- und Prüfanforderungen:

- Prüfen Sie, wie gut das Tool Ihre Compliance-Anforderungen unterstützt, einschließlich Prüfprotokollen und Berichten.
- Suchen Sie nach Funktionen, die Ihnen helfen, die Einhaltung der Vorschriften aufrechtzuerhalten und nachzuweisen.

> **Bewährte Methode**
>
> Wählen Sie ein Tool, das umfassende Auditprotokolle bereitstellt und die Erstellung von Compliance-Berichten unterstützt.

Beurteilen Sie die Rollback- und Disaster-Recovery-Funktionen:

- Beurteilen Sie die Benutzerfreundlichkeit und Zuverlässigkeit von Rollback-Mechanismen.
- Überlegen Sie, wie das Tool Notfallwiederherstellungsszenarien unterstützt.

Bewährte Methode

Testen Sie die Rollback- und Wiederherstellungsprozesse im Rahmen Ihrer Bewertung gründlich.

Häufig gestellte Fragen

F: Was sind die beliebtesten GitOps Tools für Amazon EKS?

A: Zu den beliebtesten GitOps Tools für Amazon EKS gehören Argo CD, Flux, Jenkins X und CI/CD. GitLab Jedes Tool hat Stärken, aber Argo CD und Flux sind besonders bekannt für ihren Kubernetes-nativen Ansatz und die starke Unterstützung durch die Community.

F: Wie verbessert sich das EKS-Clustermanagement? GitOps

A: GitOps Verbessert die EKS-Clusterverwaltung durch Versionskontrolle für die Infrastruktur, automatisierte Bereitstellungen, verbesserte Sicherheit durch deklarative Konfigurationen, einfachere Rollbacks und bessere Überprüfbarkeit. Es verbessert auch die Zusammenarbeit und reduziert menschliche Fehler bei Bereitstellungen.

F: Auf welche Hauptfunktionen sollte ich bei einem GitOps Tool für Amazon EKS achten?

A: Zu den wichtigsten Funktionen, auf die Sie achten sollten, gehören: nahtlose Amazon EKS-Integration, robustes RBAC, Multi-Cluster-Unterstützung, Observability-Funktionen, Unterstützung für progressive Bereitstellungsstrategien, Skalierbarkeit und Integration mit AWS-Services IAM und Amazon ECR.

F: Wie stelle ich die Sicherheit bei der Implementierung GitOps in Amazon EKS sicher?

A: Um die Sicherheit zu gewährleisten, sollten Sie ein Tool wählen, das über eine starke RBAC-Integration mit IAM, eine sichere Verwaltung von Geheimnissen, Unterstützung für verschlüsselte Git-Repositorys und die Möglichkeit verfügt, Sicherheitsrichtlinien als Code zu implementieren. Stellen Sie außerdem sicher, dass das Tool umfassende Auditprotokolle bereitstellt.

F: Können GitOps Tools Amazon EKS-Umgebungen mit mehreren Clustern verarbeiten?

A: Ja, GitOps Tools wie Argo CD und Flux verfügen über robuste Managementfunktionen für mehrere Cluster. Sie ermöglichen es Ihnen, mehrere EKS-Cluster von einer einzigen Steuerungsebene aus zu verwalten, wodurch die Konsistenz zwischen den Umgebungen gewährleistet wird.

F: Wie lassen sich GitOps Tools in bestehende CI/CD-Pipelines integrieren?

A: GitOps Tools lassen sich in der Regel in bestehende CI/CD-Pipelines integrieren, indem sie als Bereitstellungsphase der Pipeline fungieren. Sie können durch CI-Tools ausgelöst

werden, wenn Änderungen in das Git-Repository übertragen werden, und sie automatisieren den Bereitstellungsprozess für EKS-Cluster.

F: Was sind die Herausforderungen bei der Implementierung GitOps in Amazon EKS?

A: Zu den häufigsten Herausforderungen gehören die sichere Verwaltung von Geheimnissen, die Sicherstellung angemessener Zugriffskontrollen, der Umgang mit zustandsbehafteten Anwendungen, der Umgang mit Abweichungen zwischen Git und Cluster-Status und die Anpassung der Team-Workflows an das GitOps Modell.

F: Wie gehen GitOps Tools mit Rollbacks in Amazon EKS um?

A: GitOps Tools behandeln Rollbacks in der Regel, indem sie zu einem früheren Commit im Git-Repository zurückkehren. Dadurch wird automatisch ein Deployment des vorherigen als funktionierend bekannten Status ausgelöst, was zu schnellen und zuverlässigen Rollbacks führt.

F: Können GitOps Tools Amazon EKS-Add-Ons und andere AWS Ressourcen verwalten?

A: Viele GitOps Tools können Amazon EKS-Add-Ons und einige AWS Ressourcen verwalten, insbesondere wenn sie mit IaC-Tools wie Terraform oder kombiniert werden. CloudFormation Der Umfang dieser Funktion kann jedoch variieren. Spezifische Informationen zu den einzelnen GitOpsTools finden Sie im Abschnitt Tools.

F: Wie unterstützen GitOps Tools die Compliance-Anforderungen in Amazon EKS?

A: GitOps Tools unterstützen die Einhaltung der Vorschriften, indem sie einen klaren Prüfpfad für alle Änderungen bereitstellen, Genehmigungsprozesse durchsetzen, Richtlinien als Code für automatisierte Konformitätsprüfungen implementieren und detaillierte Protokollierungs- und Berichtsfunktionen bieten.

F: Was ist die Lernkurve für die Implementierung GitOps in Amazon EKS?

A: Die Lernkurve kann je nach Tool und vorhandenem Wissen Ihres Teams variieren. Im Allgemeinen passen sich Teams, die mit Git, Kubernetes und Amazon EKS vertraut sind, schneller an als andere. Die meisten gängigen Tools bieten umfangreiche Dokumentations- und Schulungsressourcen, um die Einführung zu erleichtern.

F: Wie handhaben GitOps Tools die Verwaltung von Geheimnissen in Amazon EKS?

A: GitOps Tools lassen sich in der Regel in externe Secrets-Management-Lösungen wie AWS Secrets Manager HashiCorp Vault integrieren. Einige Tools bieten auch eine integrierte Verschlüsselung für Geheimnisse, die in Git-Repositorys gespeichert sind.

F: Können GitOps Tools sowohl mit statusfreien als auch mit statusbehafteten Anwendungen in Amazon EKS funktionieren?

A: Ja, GitOps Tools können sowohl mit statusfreien als auch mit statusbehafteten Anwendungen verwendet werden. Die Verwaltung von statusbehafteten Anwendungen erfordert jedoch häufig zusätzliche Überlegungen, wie z. B. den Umgang mit persistenten Volumes und die Sicherstellung der Datenkonsistenz bei Aktualisierungen.

F: Wie unterstützen GitOps Tools Canary oder blue/green Bereitstellungen in Amazon EKS?

A: Viele GitOps Tools bieten integrierten Support für erweiterte Bereitstellungsstrategien. Sie können die schrittweise Einführung neuer Versionen verwalten, nach Problemen suchen und bei Erkennung von Problemen automatisch ein Rollback durchführen. Alle diese Operationen sind als Code im Git-Repository definiert.

F: Was ist der Unterschied zwischen der Verwendung eines GitOps Tools und der Verwendung **`kubectl apply`** mit einer CI/CD Pipeline?

A: GitOps Tools bieten Vorteile gegenüber einfachen `kubectl apply` Befehlen, darunter automatische Erkennung und Abstimmung von Abweichungen, erhöhte Sicherheit durch Pull-basierte Bereitstellungen, bessere Überprüfbarkeit und ausgefeiltere Bereitstellungsstrategien. Sie bieten auch einen umfassenderen Ansatz zur Verwaltung des gesamten Clusterstatus.

Ressourcen

Die folgenden Ressourcen bieten offizielle Dokumentation, praktische Leitfäden, Fallstudien und eingehende Analysen, die Ihnen helfen können, bei der Auswahl eines GitOps Tools für Ihren EKS-Cluster eine fundierte Entscheidung zu treffen. Sie behandeln verschiedene Aspekte von GitOps, darunter Implementierungsstrategien, bewährte Verfahren, Vergleiche zwischen verschiedenen Tools und praktische Erfahrungen.

AWS Ressourcen:

- Amazon EKS-Dokumentation
- Automatisieren von Amazon EKS mit GitOps (AWS Blogbeitrag)
- Einführung in EKS mit Weaveworks (Workshop) GitOps AWS
- Flux-Labor (Amazon EKS-Workshop)
- Argo CD-Labor (Amazon EKS-Werkstatt)

GitOps und Dokumentation zum Tool:

- GitOps Bewährte Methoden für kontinuierliche Bereitstellung und fortschrittliche Sicherheit (On-Demand-Webinar DevOps auf Abruf)
- Kubernetes-Dokumentation
- Argo-CD-Dokumentation
- Flux-Dokumentation
- Dokumentation weben GitOps
- Jenkins X-Dokumentation
- GitLab CI/CD Dokumentation
- Spinnaker-Dokumentation
- Dokumentation zur Rancher Fleet
- Codefresh-Dokumentation

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
Erste Veröffentlichung	—	30. April 2025

AWS Glossar zu präskriptiven Leitlinien

Die folgenden Begriffe werden häufig in Strategien, Leitfäden und Mustern von AWS Prescriptive Guidance verwendet. Um Einträge vorzuschlagen, verwenden Sie bitte den Link Feedback geben am Ende des Glossars.

Zahlen

7 Rs

Sieben gängige Migrationsstrategien für die Verlagerung von Anwendungen in die Cloud. Diese Strategien bauen auf den 5 Rs auf, die Gartner 2011 identifiziert hat, und bestehen aus folgenden Elementen:

- Faktorwechsel/Architekturwechsel – Verschieben Sie eine Anwendung und ändern Sie ihre Architektur, indem Sie alle Vorteile cloudnativer Feature nutzen, um Agilität, Leistung und Skalierbarkeit zu verbessern. Dies beinhaltet in der Regel die Portierung des Betriebssystems und der Datenbank. Beispiel: Migrieren Sie Ihre lokale Oracle-Datenbank auf die Amazon Aurora PostgreSQL-kompatible Edition.
- Plattformwechsel (Lift and Reshape) – Verschieben Sie eine Anwendung in die Cloud und führen Sie ein gewisses Maß an Optimierung ein, um die Cloud-Funktionen zu nutzen. Beispiel: Migrieren Sie Ihre lokale Oracle-Datenbank zu Amazon Relational Database Service (Amazon RDS) für Oracle in der. AWS Cloud
- Neukauf (Drop and Shop) – Wechseln Sie zu einem anderen Produkt, indem Sie typischerweise von einer herkömmlichen Lizenz zu einem SaaS-Modell wechseln. Beispiel: Migrieren Sie Ihr CRM-System (Customer Relationship Management) zu Salesforce.com.
- Hostwechsel (Lift and Shift) – Verschieben Sie eine Anwendung in die Cloud, ohne Änderungen vorzunehmen, um die Cloud-Funktionen zu nutzen. Beispiel: Migrieren Sie Ihre lokale Oracle-Datenbank zu Oracle auf einer EC2-Instanz in der. AWS Cloud
- Verschieben (Lift and Shift auf Hypervisor-Ebene) – Verlagern Sie die Infrastruktur in die Cloud, ohne neue Hardware kaufen, Anwendungen umschreiben oder Ihre bestehenden Abläufe ändern zu müssen. Sie migrieren Server von einer lokalen Plattform zu einem Cloud-Dienst für dieselbe Plattform. Beispiel: Migrieren Sie eine Microsoft Hyper-V Anwendung zu AWS.
- Beibehaltung (Wiederaufgreifen) – Bewahren Sie Anwendungen in Ihrer Quellumgebung auf. Dazu können Anwendungen gehören, die einen umfangreichen Faktorwechsel erfordern und

die Sie auf einen späteren Zeitpunkt verschieben möchten, sowie ältere Anwendungen, die Sie beibehalten möchten, da es keine geschäftliche Rechtfertigung für ihre Migration gibt.

- Außerbetriebnahme – Dekommissionierung oder Entfernung von Anwendungen, die in Ihrer Quellumgebung nicht mehr benötigt werden.

A

ABAC

Siehe attributbasierte Zugriffskontrolle.

abstrahierte Dienste

Siehe Managed Services.

ACID

Siehe Atomarität, Konsistenz, Isolierung und Haltbarkeit.

Aktiv-Aktiv-Migration

Eine Datenbankmigrationsmethode, bei der die Quell- und Zieldatenbanken synchron gehalten werden (mithilfe eines bidirektionalen Replikationstools oder dualer Schreibvorgänge) und beide Datenbanken Transaktionen von miteinander verbundenen Anwendungen während der Migration verarbeiten. Diese Methode unterstützt die Migration in kleinen, kontrollierten Batches, anstatt einen einmaligen Cutover zu erfordern. Es ist flexibler, erfordert aber mehr Arbeit als eine aktiv-passive Migration.

Aktiv-Passiv-Migration

Eine Datenbankmigrationsmethode, bei der die Quell- und Zieldatenbanken synchron gehalten werden, aber nur die Quelldatenbank verarbeitet Transaktionen von verbindenden Anwendungen, während Daten in die Zieldatenbank repliziert werden. Die Zieldatenbank akzeptiert während der Migration keine Transaktionen.

Aggregatfunktion

Eine SQL-Funktion, die mit einer Gruppe von Zeilen arbeitet und einen einzelnen Rückgabewert für die Gruppe berechnet. Beispiele für Aggregatfunktionen sind `SUM` und`MAX`.

AI

Siehe künstliche Intelligenz.

AIOps

Siehe Operationen im Bereich künstliche Intelligenz.

Anonymisierung

Der Prozess des dauerhaften Löschens personenbezogener Daten in einem Datensatz. Anonymisierung kann zum Schutz der Privatsphäre beitragen. Anonymisierte Daten gelten nicht mehr als personenbezogene Daten.

Anti-Muster

Eine häufig verwendete Lösung für ein wiederkehrendes Problem, bei dem die Lösung kontraproduktiv, ineffektiv oder weniger wirksam als eine Alternative ist.

Anwendungssteuerung

Ein Sicherheitsansatz, bei dem nur zugelassene Anwendungen verwendet werden können, um ein System vor Schadsoftware zu schützen.

Anwendungsportfolio

Eine Sammlung detaillierter Informationen zu jeder Anwendung, die von einer Organisation verwendet wird, einschließlich der Kosten für die Erstellung und Wartung der Anwendung und ihres Geschäftswerts. Diese Informationen sind entscheidend für den Prozess der Portfoliofindung und -analyse und hilft bei der Identifizierung und Priorisierung der Anwendungen, die migriert, modernisiert und optimiert werden sollen.

künstliche Intelligenz (KI)

Das Gebiet der Datenverarbeitungswissenschaft, das sich der Nutzung von Computertechnologien zur Ausführung kognitiver Funktionen widmet, die typischerweise mit Menschen in Verbindung gebracht werden, wie Lernen, Problemlösen und Erkennen von Mustern. Weitere Informationen finden Sie unter Was ist künstliche Intelligenz?

Operationen mit künstlicher Intelligenz (AIOps)

Der Prozess des Einsatzes von Techniken des Machine Learning zur Lösung betrieblicher Probleme, zur Reduzierung betrieblicher Zwischenfälle und menschlicher Eingriffe sowie zur Steigerung der Servicequalität. Weitere Informationen zur Verwendung in der AWS Migrationsstrategie finden Sie im Operations Integration Guide. AIOps

Asymmetrische Verschlüsselung

Ein Verschlüsselungsalgorithmus, der ein Schlüsselpaar, einen öffentlichen Schlüssel für die Verschlüsselung und einen privaten Schlüssel für die Entschlüsselung verwendet. Sie können den

öffentlichen Schlüssel teilen, da er nicht für die Entschlüsselung verwendet wird. Der Zugriff auf den privaten Schlüssel sollte jedoch stark eingeschränkt sein.

Atomizität, Konsistenz, Isolierung, Haltbarkeit (ACID)

Eine Reihe von Softwareeigenschaften, die die Datenvalidität und betriebliche Zuverlässigkeit einer Datenbank auch bei Fehlern, Stromausfällen oder anderen Problemen gewährleisten.

Attributbasierte Zugriffskontrolle (ABAC)

Die Praxis, detaillierte Berechtigungen auf der Grundlage von Benutzerattributen wie Abteilung, Aufgabenrolle und Teamname zu erstellen. Weitere Informationen finden Sie unter ABAC AWS in der AWS Identity and Access Management (IAM-) Dokumentation.

autoritative Datenquelle

Ein Ort, an dem Sie die primäre Version der Daten speichern, die als die zuverlässigste Informationsquelle angesehen wird. Sie können Daten aus der maßgeblichen Datenquelle an andere Speicherorte kopieren, um die Daten zu verarbeiten oder zu ändern, z. B. zu anonymisieren, zu redigieren oder zu pseudonymisieren.

Availability Zone

Ein bestimmter Standort innerhalb einer AWS-Region , der vor Ausfällen in anderen Availability Zones geschützt ist und kostengünstige Netzwerkkonnektivität mit niedriger Latenz zu anderen Availability Zones in derselben Region bietet.

AWS Framework für die Einführung der Cloud (AWS CAF)

Ein Framework mit Richtlinien und bewährten Verfahren, das Unternehmen bei der Entwicklung eines effizienten und effektiven Plans für die erfolgreiche Umstellung auf die Cloud unterstützt. AWS AWS CAF unterteilt die Leitlinien in sechs Schwerpunktbereiche, die als Perspektiven bezeichnet werden: Unternehmen, Mitarbeiter, Unternehmensführung, Plattform, Sicherheit und Betrieb. Die Perspektiven Geschäft, Mitarbeiter und Unternehmensführung konzentrieren sich auf Geschäftskompetenzen und -prozesse, während sich die Perspektiven Plattform, Sicherheit und Betriebsabläufe auf technische Fähigkeiten und Prozesse konzentrieren. Die Personalperspektive zielt beispielsweise auf Stakeholder ab, die sich mit Personalwesen (HR), Personalfunktionen und Personalmanagement befassen. Aus dieser Perspektive bietet AWS CAF Leitlinien für Personalentwicklung, Schulung und Kommunikation, um das Unternehmen auf eine erfolgreiche Cloud-Einführung vorzubereiten. Weitere Informationen finden Sie auf der AWS -CAF-Webseite und dem AWS -CAF-Whitepaper.

AWS Workload-Qualifizierungsrahmen (AWS WQF)

Ein Tool, das Workloads bei der Datenbankmigration bewertet, Migrationsstrategien empfiehlt und Arbeitsschätzungen bereitstellt. AWS WQF ist in () enthalten. AWS Schema Conversion Tool AWS SCT Es analysiert Datenbankschemas und Codeobjekte, Anwendungscode, Abhängigkeiten und Leistungsmerkmale und stellt Bewertungsberichte bereit.

B

schlechter Bot

Ein Bot, der Einzelpersonen oder Organisationen stören oder ihnen Schaden zufügen soll.

BCP

Siehe Planung der Geschäftskontinuität.

Verhaltensdiagramm

Eine einheitliche, interaktive Ansicht des Ressourcenverhaltens und der Interaktionen im Laufe der Zeit. Sie können ein Verhaltensdiagramm mit Amazon Detective verwenden, um fehlgeschlagene Anmeldeversuche, verdächtige API-Aufrufe und ähnliche Vorgänge zu untersuchen. Weitere Informationen finden Sie unter Daten in einem Verhaltensdiagramm in der Detective-Dokumentation.

Big-Endian-System

Ein System, welches das höchstwertige Byte zuerst speichert. Siehe auch Endianness.

Binäre Klassifikation

Ein Prozess, der ein binäres Ergebnis vorhersagt (eine von zwei möglichen Klassen). Beispielsweise könnte Ihr ML-Modell möglicherweise Probleme wie „Handelt es sich bei dieser E-Mail um Spam oder nicht?“ vorhersagen müssen oder „Ist dieses Produkt ein Buch oder ein Auto?“

Bloom-Filter

Eine probabilistische, speichereffiziente Datenstruktur, mit der getestet wird, ob ein Element Teil einer Menge ist.

Blau/Grün-Bereitstellung

Eine Bereitstellungsstrategie, bei der Sie zwei separate, aber identische Umgebungen erstellen. Sie führen die aktuelle Anwendungsversion in einer Umgebung (blau) und die neue

Anwendungsversion in der anderen Umgebung (grün) aus. Mit dieser Strategie können Sie schnell und mit minimalen Auswirkungen ein Rollback durchführen.

Bot

Eine Softwareanwendung, die automatisierte Aufgaben über das Internet ausführt und menschliche Aktivitäten oder Interaktionen simuliert. Manche Bots sind nützlich oder nützlich, wie z. B. Webcrawler, die Informationen im Internet indexieren. Einige andere Bots, sogenannte bösartige Bots, sollen Einzelpersonen oder Organisationen stören oder ihnen Schaden zufügen.

Botnetz

Netzwerke von Bots, die mit Malware infiziert sind und unter der Kontrolle einer einzigen Partei stehen, die als Bot-Herder oder Bot-Operator bezeichnet wird. Botnetze sind der bekannteste Mechanismus zur Skalierung von Bots und ihrer Wirkung.

branch

Ein containerisierter Bereich eines Code-Repositorys. Der erste Zweig, der in einem Repository erstellt wurde, ist der Hauptzweig. Sie können einen neuen Zweig aus einem vorhandenen Zweig erstellen und dann Feature entwickeln oder Fehler in dem neuen Zweig beheben. Ein Zweig, den Sie erstellen, um ein Feature zu erstellen, wird allgemein als Feature-Zweig bezeichnet. Wenn das Feature zur Veröffentlichung bereit ist, führen Sie den Feature-Zweig wieder mit dem Hauptzweig zusammen. Weitere Informationen finden Sie unter Über Branches (GitHub Dokumentation).

Zugang durch Glasbruch

Unter außergewöhnlichen Umständen und im Rahmen eines genehmigten Verfahrens ist dies eine schnelle Methode für einen Benutzer, auf einen Bereich zuzugreifen AWS-Konto , für den er normalerweise keine Zugriffsrechte besitzt. Weitere Informationen finden Sie unter dem Indikator Implementation break-glass procedures in den AWS Well-Architected-Leitlinien.

Brownfield-Strategie

Die bestehende Infrastruktur in Ihrer Umgebung. Wenn Sie eine Brownfield-Strategie für eine Systemarchitektur anwenden, richten Sie sich bei der Gestaltung der Architektur nach den Einschränkungen der aktuellen Systeme und Infrastruktur. Wenn Sie die bestehende Infrastruktur erweitern, könnten Sie Brownfield- und Greenfield-Strategien mischen.

Puffer-Cache

Der Speicherbereich, in dem die am häufigsten abgerufenen Daten gespeichert werden.

Geschäftsfähigkeit

Was ein Unternehmen tut, um Wert zu generieren (z. B. Vertrieb, Kundenservice oder Marketing). Microservices-Architekturen und Entwicklungsentscheidungen können von den Geschäftskapazitäten beeinflusst werden. Weitere Informationen finden Sie im Abschnitt Organisiert nach Geschäftskapazitäten des Whitepapers Ausführen von containerisierten Microservices in AWS.

Planung der Geschäftskontinuität (BCP)

Ein Plan, der die potenziellen Auswirkungen eines störenden Ereignisses, wie z. B. einer groß angelegten Migration, auf den Betrieb berücksichtigt und es einem Unternehmen ermöglicht, den Betrieb schnell wieder aufzunehmen.

C

CAF

Weitere Informationen finden Sie unter Framework AWS für die Cloud-Einführung.

Bereitstellung auf Kanaren

Die langsame und schrittweise Veröffentlichung einer Version für Endbenutzer. Wenn Sie sich sicher sind, stellen Sie die neue Version bereit und ersetzen die aktuelle Version vollständig.

CCoE

Weitere Informationen finden Sie im Cloud Center of Excellence.

CDC

Siehe Erfassung von Änderungsdaten.

Erfassung von Datenänderungen (CDC)

Der Prozess der Nachverfolgung von Änderungen an einer Datenquelle, z. B. einer Datenbanktabelle, und der Aufzeichnung von Metadaten zu der Änderung. Sie können CDC für verschiedene Zwecke verwenden, z. B. für die Prüfung oder Replikation von Änderungen in einem Zielsystem, um die Synchronisation aufrechtzuerhalten.

Chaos-Technik

Absichtliches Einführen von Ausfällen oder Störungsereignissen, um die Widerstandsfähigkeit eines Systems zu testen. Sie können AWS Fault Injection Service (AWS FIS) verwenden, um Experimente durchzuführen, die Ihre AWS Workloads stress, und deren Reaktion zu bewerten.

CI/CD

Siehe Continuous Integration und Continuous Delivery.

Klassifizierung

Ein Kategorisierungsprozess, der bei der Erstellung von Vorhersagen hilft. ML-Modelle für Klassifikationsprobleme sagen einen diskreten Wert voraus. Diskrete Werte unterscheiden sich immer voneinander. Beispielsweise muss ein Modell möglicherweise auswerten, ob auf einem Bild ein Auto zu sehen ist oder nicht.

clientseitige Verschlüsselung

Lokale Verschlüsselung von Daten, bevor das Ziel sie AWS-Service empfängt.

Cloud-Exzellenzzentrum (CCoE)

Ein multidisziplinäres Team, das die Cloud-Einführung in der gesamten Organisation vorantreibt, einschließlich der Entwicklung bewährter Cloud-Methoden, der Mobilisierung von Ressourcen, der Festlegung von Migrationszeitplänen und der Begleitung der Organisation durch groß angelegte Transformationen. Weitere Informationen finden Sie in den CCoE-Beiträgen im AWS Cloud Enterprise Strategy Blog.

Cloud Computing

Die Cloud-Technologie, die typischerweise für die Ferndatenspeicherung und das IoT-Gerätemanagement verwendet wird. Cloud Computing ist häufig mit Edge-Computing-Technologie verbunden.

Cloud-Betriebsmodell

In einer IT-Organisation das Betriebsmodell, das zum Aufbau, zur Weiterentwicklung und Optimierung einer oder mehrerer Cloud-Umgebungen verwendet wird. Weitere Informationen finden Sie unter Aufbau Ihres Cloud-Betriebsmodells.

Phasen der Einführung der Cloud

Die vier Phasen, die Unternehmen bei der Migration in der Regel durchlaufen AWS Cloud:

- Projekt – Durchführung einiger Cloud-bezogener Projekte zu Machbarkeitsnachweisen und zu Lernzwecken
- Fundament — Tätigen Sie grundlegende Investitionen, um Ihre Cloud-Einführung zu skalieren (z. B. Einrichtung einer landing zone, Definition eines CCo E, Einrichtung eines Betriebsmodells)

- Migration – Migrieren einzelner Anwendungen
- Neuentwicklung – Optimierung von Produkten und Services und Innovation in der Cloud

Diese Phasen wurden von Stephen Orban im Blogbeitrag The Journey Toward Cloud-First & the Stages of Adoption im AWS Cloud Enterprise Strategy-Blog definiert. Informationen darüber, wie sie mit der AWS Migrationsstrategie zusammenhängen, finden Sie im Leitfaden zur Vorbereitung der Migration.

CMDB

Siehe Datenbank für das Konfigurationsmanagement.

Code-Repository

Ein Ort, an dem Quellcode und andere Komponenten wie Dokumentation, Beispiele und Skripts gespeichert und im Rahmen von Versionskontrollprozessen aktualisiert werden. Zu den gängigen Cloud-Repositorys gehören GitHub oderBitbucket Cloud. Jede Version des Codes wird Zweig genannt. In einer Microservice-Struktur ist jedes Repository einer einzelnen Funktionalität gewidmet. Eine einzelne CI/CD-Pipeline kann mehrere Repositorien verwenden.

Kalter Cache

Ein Puffer-Cache, der leer oder nicht gut gefüllt ist oder veraltete oder irrelevante Daten enthält. Dies beeinträchtigt die Leistung, da die Datenbank-Instance aus dem Hauptspeicher oder der Festplatte lesen muss, was langsamer ist als das Lesen aus dem Puffercache.

Kalte Daten

Daten, auf die selten zugegriffen wird und die in der Regel historisch sind. Bei der Abfrage dieser Art von Daten sind langsame Abfragen in der Regel akzeptabel. Durch die Verlagerung dieser Daten auf leistungsschwächere und kostengünstigere Speicherstufen oder -klassen können Kosten gesenkt werden.

Computer Vision (CV)

Ein Bereich der KI, der maschinelles Lernen nutzt, um Informationen aus visuellen Formaten wie digitalen Bildern und Videos zu analysieren und zu extrahieren. Amazon SageMaker AI bietet beispielsweise Bildverarbeitungsalgorithmen für CV.

Drift in der Konfiguration

Bei einer Arbeitslast eine Änderung der Konfiguration gegenüber dem erwarteten Zustand. Dies kann dazu führen, dass der Workload nicht mehr richtlinienkonform wird, und zwar in der Regel schrittweise und unbeabsichtigt.

Verwaltung der Datenbankkonfiguration (CMDB)

Ein Repository, das Informationen über eine Datenbank und ihre IT-Umgebung speichert und verwaltet, inklusive Hardware- und Softwarekomponenten und deren Konfigurationen. In der Regel verwenden Sie Daten aus einer CMDB in der Phase der Portfolioerkennung und -analyse der Migration.

Konformitätspaket

Eine Sammlung von AWS Config Regeln und Abhilfemaßnahmen, die Sie zusammenstellen können, um Ihre Konformitäts- und Sicherheitsprüfungen individuell anzupassen. Mithilfe einer YAML-Vorlage können Sie ein Conformance Pack als einzelne Entität in einer AWS-Konto AND-Region oder unternehmensweit bereitstellen. Weitere Informationen finden Sie in der Dokumentation unter Conformance Packs. AWS Config

Kontinuierliche Bereitstellung und kontinuierliche Integration (CI/CD)

Der Prozess der Automatisierung der Quell-, Build-, Test-, Staging- und Produktionsphasen des Softwareveröffentlichungsprozesses. CI/CD wird allgemein als Pipeline beschrieben. CI/CD kann Ihnen helfen, Prozesse zu automatisieren, die Produktivität zu steigern, die Codequalität zu verbessern und schneller zu liefern. Weitere Informationen finden Sie unter Vorteile der kontinuierlichen Auslieferung. CD kann auch für kontinuierliche Bereitstellung stehen. Weitere Informationen finden Sie unterKontinuierliche Auslieferung im Vergleich zu kontinuierlicher Bereitstellung.

CV

Siehe Computer Vision.

D

Daten im Ruhezustand

Daten, die in Ihrem Netzwerk stationär sind, z. B. Daten, die sich im Speicher befinden.

Datenklassifizierung

Ein Prozess zur Identifizierung und Kategorisierung der Daten in Ihrem Netzwerk auf der Grundlage ihrer Kritikalität und Sensitivität. Sie ist eine wichtige Komponente jeder Strategie für das Management von Cybersecurity-Risiken, da sie Ihnen hilft, die geeigneten Schutz- und Aufbewahrungskontrollen für die Daten zu bestimmen. Die Datenklassifizierung ist ein Bestandteil

der Sicherheitssäule im AWS Well-Architected Framework. Weitere Informationen finden Sie unter Datenklassifizierung.

Datendrift

Eine signifikante Abweichung zwischen den Produktionsdaten und den Daten, die zum Trainieren eines ML-Modells verwendet wurden, oder eine signifikante Änderung der Eingabedaten im Laufe der Zeit. Datendrift kann die Gesamtqualität, Genauigkeit und Fairness von ML-Modellvorhersagen beeinträchtigen.

Daten während der Übertragung

Daten, die sich aktiv durch Ihr Netzwerk bewegen, z. B. zwischen Netzwerkressourcen.

Datennetz

Ein architektonisches Framework, das verteilte, dezentrale Dateneigentum mit zentraler Verwaltung und Steuerung ermöglicht.

Datenminimierung

Das Prinzip, nur die Daten zu sammeln und zu verarbeiten, die unbedingt erforderlich sind. Durch Datenminimierung im AWS Cloud können Datenschutzrisiken, Kosten und der CO2-Fußabdruck Ihrer Analysen reduziert werden.

Datenperimeter

Eine Reihe präventiver Schutzmaßnahmen in Ihrer AWS Umgebung, die sicherstellen, dass nur vertrauenswürdige Identitäten auf vertrauenswürdige Ressourcen von erwarteten Netzwerken zugreifen. Weitere Informationen finden Sie unter Aufbau eines Datenperimeters auf. AWS

Vorverarbeitung der Daten

Rohdaten in ein Format umzuwandeln, das von Ihrem ML-Modell problemlos verarbeitet werden kann. Die Vorverarbeitung von Daten kann bedeuten, dass bestimmte Spalten oder Zeilen entfernt und fehlende, inkonsistente oder doppelte Werte behoben werden.

Herkunft der Daten

Der Prozess der Nachverfolgung des Ursprungs und der Geschichte von Daten während ihres gesamten Lebenszyklus, z. B. wie die Daten generiert, übertragen und gespeichert wurden.

betroffene Person

Eine Person, deren Daten gesammelt und verarbeitet werden.

Data Warehouse

Ein Datenverwaltungssystem, das Business Intelligence wie Analysen unterstützt. Data Warehouses enthalten in der Regel große Mengen historischer Daten und werden in der Regel für Abfragen und Analysen verwendet.

Datenbankdefinitionssprache (DDL)

Anweisungen oder Befehle zum Erstellen oder Ändern der Struktur von Tabellen und Objekten in einer Datenbank.

Datenbankmanipulationssprache (DML)

Anweisungen oder Befehle zum Ändern (Einfügen, Aktualisieren und Löschen) von Informationen in einer Datenbank.

DDL

Siehe Datenbankdefinitionssprache.

Deep-Ensemble

Mehrere Deep-Learning-Modelle zur Vorhersage kombinieren. Sie können Deep-Ensembles verwenden, um eine genauere Vorhersage zu erhalten oder um die Unsicherheit von Vorhersagen abzuschätzen.

Deep Learning

Ein ML-Teilbereich, der mehrere Schichten künstlicher neuronaler Netzwerke verwendet, um die Zuordnung zwischen Eingabedaten und Zielvariablen von Interesse zu ermitteln.

defense-in-depth

Ein Ansatz zur Informationssicherheit, bei dem eine Reihe von Sicherheitsmechanismen und -kontrollen sorgfältig in einem Computernetzwerk verteilt werden, um die Vertraulichkeit, Integrität und Verfügbarkeit des Netzwerks und der darin enthaltenen Daten zu schützen. Wenn Sie diese Strategie anwenden AWS, fügen Sie mehrere Steuerelemente auf verschiedenen Ebenen der AWS Organizations Struktur hinzu, um die Ressourcen zu schützen. Ein defense-in-depth Ansatz könnte beispielsweise Multi-Faktor-Authentifizierung, Netzwerksegmentierung und Verschlüsselung kombinieren.

delegierter Administrator

In AWS Organizations kann ein kompatibler Dienst ein AWS Mitgliedskonto registrieren, um die Konten der Organisation und die Berechtigungen für diesen Dienst zu verwalten. Dieses Konto

wird als delegierter Administrator für diesen Service bezeichnet. Weitere Informationen und eine Liste kompatibler Services finden Sie unter Services, die mit AWS Organizations funktionieren in der AWS Organizations -Dokumentation.

Einsatz

Der Prozess, bei dem eine Anwendung, neue Feature oder Codekorrekturen in der Zielumgebung verfügbar gemacht werden. Die Bereitstellung umfasst das Implementieren von Änderungen an einer Codebasis und das anschließende Erstellen und Ausführen dieser Codebasis in den Anwendungsumgebungen.

Entwicklungsumgebung

Siehe Umgebung.

Detektivische Kontrolle

Eine Sicherheitskontrolle, die darauf ausgelegt ist, ein Ereignis zu erkennen, zu protokollieren und zu warnen, nachdem ein Ereignis eingetreten ist. Diese Kontrollen stellen eine zweite Verteidigungslinie dar und warnen Sie vor Sicherheitsereignissen, bei denen die vorhandenen präventiven Kontrollen umgangen wurden. Weitere Informationen finden Sie unter Detektivische Kontrolle in Implementierung von Sicherheitskontrollen in AWS.

Abbildung des Wertstroms in der Entwicklung (DVSM)

Ein Prozess zur Identifizierung und Priorisierung von Einschränkungen, die sich negativ auf Geschwindigkeit und Qualität im Lebenszyklus der Softwareentwicklung auswirken. DVSM erweitert den Prozess der Wertstromanalyse, der ursprünglich für Lean-Manufacturing-Praktiken konzipiert wurde. Es konzentriert sich auf die Schritte und Teams, die erforderlich sind, um durch den Softwareentwicklungsprozess Mehrwert zu schaffen und zu steigern.

digitaler Zwilling

Eine virtuelle Darstellung eines realen Systems, z. B. eines Gebäudes, einer Fabrik, einer Industrieanlage oder einer Produktionslinie. Digitale Zwillinge unterstützen vorausschauende Wartung, Fernüberwachung und Produktionsoptimierung.

Maßtabelle

In einem Sternschema eine kleinere Tabelle, die Datenattribute zu quantitativen Daten in einer Faktentabelle enthält. Bei Attributen von Dimensionstabellen handelt es sich in der Regel um Textfelder oder diskrete Zahlen, die sich wie Text verhalten. Diese Attribute werden häufig zum Einschränken von Abfragen, zum Filtern und zur Kennzeichnung von Ergebnismengen verwendet.

Katastrophe

Ein Ereignis, das verhindert, dass ein Workload oder ein System seine Geschäftsziele an seinem primären Einsatzort erfüllt. Diese Ereignisse können Naturkatastrophen, technische Ausfälle oder das Ergebnis menschlichen Handelns sein, wie z. B. unbeabsichtigte Fehlkonfigurationen oder ein Malware-Angriff.

Notfallwiederherstellung (DR)

Die Strategie und der Prozess, mit denen Sie Ausfallzeiten und Datenverluste aufgrund einer Katastrophe minimieren. Weitere Informationen finden Sie unter Disaster Recovery von Workloads unter AWS: Wiederherstellung in der Cloud im AWS Well-Architected Framework.

DML

Siehe Sprache zur Datenbankmanipulation.

Domainorientiertes Design

Ein Ansatz zur Entwicklung eines komplexen Softwaresystems, bei dem seine Komponenten mit sich entwickelnden Domains oder Kerngeschäftszielen verknüpft werden, denen jede Komponente dient. Dieses Konzept wurde von Eric Evans in seinem Buch Domaingesteuertes Design: Bewältigen der Komplexität im Herzen der Software (Boston: Addison-Wesley Professional, 2003) vorgestellt. Informationen darüber, wie Sie domaingesteuertes Design mit dem Strangler-Fig-Muster verwenden können, finden Sie unter Schrittweises Modernisieren älterer Microsoft ASP.NET (ASMX)-Webservices mithilfe von Containern und Amazon API Gateway.

DR

Siehe Disaster Recovery.

Erkennung von Driften

Verfolgung von Abweichungen von einer Basiskonfiguration. Sie können es beispielsweise verwenden, AWS CloudFormation um Abweichungen bei den Systemressourcen zu erkennen, oder Sie können AWS Control Tower damit Änderungen in Ihrer landing zone erkennen, die sich auf die Einhaltung von Governance-Anforderungen auswirken könnten.

DVSM

Siehe Abbildung des Wertstroms in der Entwicklung.

E

EDA

Siehe explorative Datenanalyse.

EDI

Siehe elektronischer Datenaustausch.

Edge-Computing

Die Technologie, die die Rechenleistung für intelligente Geräte an den Rändern eines IoT-Netzwerks erhöht. Im Vergleich zu Cloud Computing kann Edge Computing die Kommunikationslatenz reduzieren und die Reaktionszeit verbessern.

elektronischer Datenaustausch (EDI)

Der automatisierte Austausch von Geschäftsdokumenten zwischen Organisationen. Weitere Informationen finden Sie unter Was ist elektronischer Datenaustausch.

Verschlüsselung

Ein Rechenprozess, der Klartextdaten, die für Menschen lesbar sind, in Chiffretext umwandelt.

Verschlüsselungsschlüssel

Eine kryptografische Zeichenfolge aus zufälligen Bits, die von einem Verschlüsselungsalgorithmus generiert wird. Schlüssel können unterschiedlich lang sein, und jeder Schlüssel ist so konzipiert, dass er unvorhersehbar und einzigartig ist.

Endianismus

Die Reihenfolge, in der Bytes im Computerspeicher gespeichert werden. Big-Endian-Systeme speichern das höchstwertige Byte zuerst. Little-Endian-Systeme speichern das niedrigwertigste Byte zuerst.

Endpunkt

Siehe Service-Endpunkt.

Endpunkt-Services

Ein Service, den Sie in einer Virtual Private Cloud (VPC) hosten können, um ihn mit anderen Benutzern zu teilen. Sie können einen Endpunktdienst mit anderen AWS-Konten oder AWS Identity and Access Management (IAM AWS PrivateLink -) Prinzipalen erstellen und diesen

Berechtigungen gewähren. Diese Konten oder Prinzipale können sich privat mit Ihrem Endpunktservice verbinden, indem sie Schnittstellen-VPC-Endpunkte erstellen. Weitere Informationen finden Sie unter Einen Endpunkt-Service erstellen in der Amazon Virtual Private Cloud (Amazon VPC)-Dokumentation.

Unternehmensressourcenplanung (ERP)

Ein System, das wichtige Geschäftsprozesse (wie Buchhaltung, MES und Projektmanagement) für ein Unternehmen automatisiert und verwaltet.

Envelope-Verschlüsselung

Der Prozess der Verschlüsselung eines Verschlüsselungsschlüssels mit einem anderen Verschlüsselungsschlüssel. Weitere Informationen finden Sie unter Envelope-Verschlüsselung in der AWS Key Management Service (AWS KMS) -Dokumentation.

Umgebung

Eine Instance einer laufenden Anwendung. Die folgenden Arten von Umgebungen sind beim Cloud-Computing üblich:

- Entwicklungsumgebung – Eine Instance einer laufenden Anwendung, die nur dem Kernteam zur Verfügung steht, das für die Wartung der Anwendung verantwortlich ist. Entwicklungsumgebungen werden verwendet, um Änderungen zu testen, bevor sie in höhere Umgebungen übertragen werden. Diese Art von Umgebung wird manchmal als Testumgebung bezeichnet.
- Niedrigere Umgebungen – Alle Entwicklungsumgebungen für eine Anwendung, z. B. solche, die für erste Builds und Tests verwendet wurden.
- Produktionsumgebung – Eine Instance einer laufenden Anwendung, auf die Endbenutzer zugreifen können. In einer CI/CD Pipeline ist die Produktionsumgebung die letzte Bereitstellungsumgebung.
- Höhere Umgebungen – Alle Umgebungen, auf die auch andere Benutzer als das Kernentwicklungsteam zugreifen können. Dies kann eine Produktionsumgebung, Vorproduktionsumgebungen und Umgebungen für Benutzerakzeptanztests umfassen.

Epics

In der agilen Methodik sind dies funktionale Kategorien, die Ihnen helfen, Ihre Arbeit zu organisieren und zu priorisieren. Epics bieten eine allgemeine Beschreibung der Anforderungen und Implementierungsaufgaben. Zu den Sicherheitsepen AWS von CAF gehören beispielsweise Identitäts- und Zugriffsmanagement, Detektivkontrollen, Infrastruktursicherheit, Datenschutz und

Reaktion auf Vorfälle. Weitere Informationen zu Epics in der AWS -Migrationsstrategie finden Sie im Leitfaden zur Programm-Implementierung.

ERP

Siehe Enterprise Resource Planning.

Explorative Datenanalyse (EDA)

Der Prozess der Analyse eines Datensatzes, um seine Hauptmerkmale zu verstehen. Sie sammeln oder aggregieren Daten und führen dann erste Untersuchungen durch, um Muster zu finden, Anomalien zu erkennen und Annahmen zu überprüfen. EDA wird durchgeführt, indem zusammenfassende Statistiken berechnet und Datenvisualisierungen erstellt werden.

F

Faktentabelle

Die zentrale Tabelle in einem Sternschema. Sie speichert quantitative Daten über den Geschäftsbetrieb. In der Regel enthält eine Faktentabelle zwei Arten von Spalten: Spalten, die Kennzahlen enthalten, und Spalten, die einen Fremdschlüssel für eine Dimensionstabelle enthalten.

schnell scheitern

Eine Philosophie, die häufige und inkrementelle Tests verwendet, um den Entwicklungslebenszyklus zu verkürzen. Dies ist ein wichtiger Bestandteil eines agilen Ansatzes.

Grenze zur Fehlerisolierung

Dabei handelt es sich um eine Grenze AWS Cloud, z. B. eine Availability Zone AWS-Region, eine Steuerungsebene oder eine Datenebene, die die Auswirkungen eines Fehlers begrenzt und die Widerstandsfähigkeit von Workloads verbessert. Weitere Informationen finden Sie unter Grenzen zur AWS Fehlerisolierung.

Feature-Zweig

Siehe Zweig.

Features

Die Eingabedaten, die Sie verwenden, um eine Vorhersage zu treffen. In einem Fertigungskontext könnten Feature beispielsweise Bilder sein, die regelmäßig von der Fertigungslinie aus aufgenommen werden.

Bedeutung der Feature

Wie wichtig ein Feature für die Vorhersagen eines Modells ist. Dies wird in der Regel als numerischer Wert ausgedrückt, der mit verschiedenen Techniken wie Shapley Additive Explanations (SHAP) und integrierten Gradienten berechnet werden kann. Weitere Informationen finden Sie unter Interpretierbarkeit von Modellen für maschinelles Lernen mit AWS.

Featuretransformation

Daten für den ML-Prozess optimieren, einschließlich der Anreicherung von Daten mit zusätzlichen Quellen, der Skalierung von Werten oder der Extraktion mehrerer Informationssätze aus einem einzigen Datenfeld. Das ermöglicht dem ML-Modell, von den Daten profitieren. Wenn Sie beispielsweise das Datum „27.05.2021 00:15:37“ in „2021“, „Mai“, „Donnerstag“ und „15“ aufschlüsseln, können Sie dem Lernalgorithmus helfen, nuancierte Muster zu erlernen, die mit verschiedenen Datenkomponenten verknüpft sind.

Eingabeaufforderung mit wenigen Klicks

Bereitstellung einer kleinen Anzahl von Beispielen, die die Aufgabe und das gewünschte Ergebnis veranschaulichen, bevor das LLM aufgefordert wird, eine ähnliche Aufgabe auszuführen. Bei dieser Technik handelt es sich um eine Anwendung des kontextbezogenen Lernens, bei der Modelle anhand von Beispielen (Aufnahmen) lernen, die in Eingabeaufforderungen eingebettet sind. Bei Aufgaben, die spezifische Formatierungs-, Argumentations- oder Fachkenntnisse erfordern, kann die Eingabeaufforderung mit wenigen Handgriffen effektiv sein. Siehe auch Zero-Shot Prompting.

FGAC

Siehe detaillierte Zugriffskontrolle.

Feinkörnige Zugriffskontrolle (FGAC)

Die Verwendung mehrerer Bedingungen, um eine Zugriffsanfrage zuzulassen oder abzulehnen.

Flash-Cut-Migration

Eine Datenbankmigrationsmethode, bei der eine kontinuierliche Datenreplikation durch Erfassung von Änderungsdaten verwendet wird, um Daten in kürzester Zeit zu migrieren, anstatt einen schrittweisen Ansatz zu verwenden. Ziel ist es, Ausfallzeiten auf ein Minimum zu beschränken.

FM

Siehe Fundamentmodell.

Fundamentmodell (FM)

Ein großes neuronales Deep-Learning-Netzwerk, das mit riesigen Datensätzen generalisierter und unbeschrifteter Daten trainiert wurde. FMs sind in der Lage, eine Vielzahl allgemeiner Aufgaben zu erfüllen, z. B. Sprache zu verstehen, Text und Bilder zu generieren und Konversationen in natürlicher Sprache zu führen. Weitere Informationen finden Sie unter Was sind Foundation-Modelle.

G

Generative KI

Eine Untergruppe von KI-Modellen, die mit großen Datenmengen trainiert wurden und mit einer einfachen Textaufforderung neue Inhalte und Artefakte wie Bilder, Videos, Text und Audio erstellen können. Weitere Informationen finden Sie unter Was ist Generative KI.

Geoblocking

Siehe geografische Einschränkungen.

Geografische Einschränkungen (Geoblocking)

Bei Amazon eine Option CloudFront, um zu verhindern, dass Benutzer in bestimmten Ländern auf Inhaltsverteilungen zugreifen. Sie können eine Zulassungsliste oder eine Sperrliste verwenden, um zugelassene und gesperrte Länder anzugeben. Weitere Informationen finden Sie in der Dokumentation unter Beschränkung der geografischen Verteilung Ihrer Inhalte. CloudFront

Gitflow-Workflow

Ein Ansatz, bei dem niedrigere und höhere Umgebungen unterschiedliche Zweige in einem Quellcode-Repository verwenden. Der Gitflow-Workflow gilt als veraltet, und der Trunk-basierte Workflow ist der moderne, bevorzugte Ansatz.

goldenes Bild

Ein Snapshot eines Systems oder einer Software, der als Vorlage für die Bereitstellung neuer Instanzen dieses Systems oder dieser Software verwendet wird. In der Fertigung kann ein Golden Image beispielsweise zur Bereitstellung von Software auf mehreren Geräten verwendet werden und trägt zur Verbesserung der Geschwindigkeit, Skalierbarkeit und Produktivität bei der Geräteherstellung bei.

Greenfield-Strategie

Das Fehlen vorhandener Infrastruktur in einer neuen Umgebung. Bei der Einführung einer Neuausrichtung einer Systemarchitektur können Sie alle neuen Technologien ohne Einschränkung der Kompatibilität mit der vorhandenen Infrastruktur auswählen, auch bekannt als Brownfield. Wenn Sie die bestehende Infrastruktur erweitern, könnten Sie Brownfield- und Greenfield-Strategien mischen.

Integritätsschutz

Eine allgemeine Regel, die dazu beiträgt, Ressourcen, Richtlinien und die Einhaltung von Vorschriften in allen Unternehmenseinheiten zu regeln (OUs). Präventiver Integritätsschutz setzt Richtlinien durch, um die Einhaltung von Standards zu gewährleisten. Sie werden mithilfe von Service-Kontrollrichtlinien und IAM-Berechtigungsgrenzen implementiert. Detektivischer Integritätsschutz erkennt Richtlinienverstöße und Compliance-Probleme und generiert Warnmeldungen zur Abhilfe. Sie werden mithilfe von AWS Config, AWS Security Hub CSPM, Amazon GuardDuty AWS Trusted Advisor, Amazon Inspector und benutzerdefinierten AWS Lambda Prüfungen implementiert.

H

HEKTAR

Siehe Hochverfügbarkeit.

Heterogene Datenbankmigration

Migrieren Sie Ihre Quelldatenbank in eine Zieldatenbank, die eine andere Datenbank-Engine verwendet (z. B. Oracle zu Amazon Aurora). Eine heterogene Migration ist in der Regel Teil einer Neuarchitektur, und die Konvertierung des Schemas kann eine komplexe Aufgabe sein. AWS bietet AWS SCT, welches bei Schemakonvertierungen hilft.

hohe Verfügbarkeit (HA)

Die Fähigkeit eines Workloads, im Falle von Herausforderungen oder Katastrophen kontinuierlich und ohne Eingreifen zu arbeiten. HA-Systeme sind so konzipiert, dass sie automatisch ein Failover durchführen, gleichbleibend hohe Leistung bieten und unterschiedliche Lasten und Ausfälle mit minimalen Leistungseinbußen bewältigen.

historische Modernisierung

Ein Ansatz zur Modernisierung und Aufrüstung von Betriebstechnologiesystemen (OT), um den Bedürfnissen der Fertigungsindustrie besser gerecht zu werden. Ein Historian ist eine Art von Datenbank, die verwendet wird, um Daten aus verschiedenen Quellen in einer Fabrik zu sammeln und zu speichern.

Daten zurückhalten

Ein Teil historischer, beschrifteter Daten, der aus einem Datensatz zurückgehalten wird, der zum Trainieren eines Modells für maschinelles Lernen verwendet wird. Sie können Holdout-Daten verwenden, um die Modellleistung zu bewerten, indem Sie die Modellvorhersagen mit den Holdout-Daten vergleichen.

Homogene Datenbankmigration

Migrieren Sie Ihre Quelldatenbank zu einer Zieldatenbank, die dieselbe Datenbank-Engine verwendet (z. B. Microsoft SQL Server zu Amazon RDS für SQL Server). Eine homogene Migration ist in der Regel Teil eines Hostwechsels oder eines Plattformwechsels. Sie können native Datenbankserviceprogramme verwenden, um das Schema zu migrieren.

heiße Daten

Daten, auf die häufig zugegriffen wird, z. B. Echtzeitdaten oder aktuelle Translationsdaten. Für diese Daten ist in der Regel eine leistungsstarke Speicherebene oder -klasse erforderlich, um schnelle Abfrageantworten zu ermöglichen.

Hotfix

Eine dringende Lösung für ein kritisches Problem in einer Produktionsumgebung. Aufgrund seiner Dringlichkeit wird ein Hotfix normalerweise außerhalb des typischen DevOps Release-Workflows erstellt.

Hypercare-Phase

Unmittelbar nach dem Cutover, der Zeitraum, in dem ein Migrationsteam die migrierten Anwendungen in der Cloud verwaltet und überwacht, um etwaige Probleme zu beheben. In der Regel dauert dieser Zeitraum 1–4 Tage. Am Ende der Hypercare-Phase überträgt das Migrationsteam in der Regel die Verantwortung für die Anwendungen an das Cloud-Betriebsteam.

I

IaC

Sehen Sie Infrastruktur als Code.

Identitätsbasierte Richtlinie

Eine Richtlinie, die einem oder mehreren IAM-Prinzipalen zugeordnet ist und deren Berechtigungen innerhalb der AWS Cloud Umgebung definiert.

Leerlaufanwendung

Eine Anwendung mit einer durchschnittlichen CPU- und Arbeitsspeicherauslastung zwischen 5 und 20 Prozent über einen Zeitraum von 90 Tagen. In einem Migrationsprojekt ist es üblich, diese Anwendungen außer Betrieb zu nehmen oder sie On-Premises beizubehalten.

IIoT

Siehe Industrielles Internet der Dinge.

unveränderliche Infrastruktur

Ein Modell, das eine neue Infrastruktur für Produktionsworkloads bereitstellt, anstatt die bestehende Infrastruktur zu aktualisieren, zu patchen oder zu modifizieren. Unveränderliche Infrastrukturen sind von Natur aus konsistenter, zuverlässiger und vorhersehbarer als veränderliche Infrastrukturen. Weitere Informationen finden Sie in der Best Practice Deploy using immutable infrastructure im AWS Well-Architected Framework.

Eingehende (ingress) VPC

In einer Architektur AWS mit mehreren Konten ist dies eine VPC, die Netzwerkverbindungen von außerhalb einer Anwendung akzeptiert, überprüft und weiterleitet. Die AWS Security Reference Architecture empfiehlt, Ihr Netzwerkkonto mit eingehendem und ausgehendem Datenverkehr und Inspektion einzurichten, VPCs um die bidirektionale Schnittstelle zwischen Ihrer Anwendung und dem Internet im weiteren Sinne zu schützen.

Inkrementelle Migration

Eine Cutover-Strategie, bei der Sie Ihre Anwendung in kleinen Teilen migrieren, anstatt eine einziges vollständiges Cutover durchzuführen. Beispielsweise könnten Sie zunächst nur einige Microservices oder Benutzer auf das neue System umstellen. Nachdem Sie sich vergewissert haben, dass alles ordnungsgemäß funktioniert, können Sie weitere Microservices oder Benutzer

schrittweise verschieben, bis Sie Ihr Legacy-System außer Betrieb nehmen können. Diese Strategie reduziert die mit großen Migrationen verbundenen Risiken.

Industrie 4.0

Ein Begriff, der 2016 von Klaus Schwab eingeführt wurde und sich auf die Modernisierung von Fertigungsprozessen durch Fortschritte in den Bereichen Konnektivität, Echtzeitdaten, Automatisierung, Analytik und KI/ML bezieht.

Infrastruktur

Alle Ressourcen und Komponenten, die in der Umgebung einer Anwendung enthalten sind.

Infrastructure as Code (IaC)

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.

industrielles Internet der Dinge (T) IIo

Einsatz von mit dem Internet verbundenen Sensoren und Geräten in Industriesektoren wie Fertigung, Energie, Automobilindustrie, Gesundheitswesen, Biowissenschaften und Landwirtschaft. Weitere Informationen finden Sie unter Aufbau einer digitalen Transformationsstrategie für das industrielle Internet der Dinge (IIoT).

Inspektions-VPC

In einer Architektur AWS mit mehreren Konten eine zentralisierte VPC, die Inspektionen des Netzwerkverkehrs zwischen VPCs (in demselben oder unterschiedlichen AWS-Regionen), dem Internet und lokalen Netzwerken verwaltet. In der AWS Security Reference Architecture wird empfohlen, Ihr Netzwerkkonto mit eingehendem und ausgehendem Datenverkehr sowie Inspektionen einzurichten, VPCs um die bidirektionale Schnittstelle zwischen Ihrer Anwendung und dem Internet im weiteren Sinne zu schützen.

Internet of Things (IoT)

Das Netzwerk verbundener physischer Objekte mit eingebetteten Sensoren oder Prozessoren, das über das Internet oder über ein lokales Kommunikationsnetzwerk mit anderen Geräten und Systemen kommuniziert. Weitere Informationen finden Sie unter Was ist IoT?

Interpretierbarkeit

Ein Merkmal eines Modells für Machine Learning, das beschreibt, inwieweit ein Mensch verstehen kann, wie die Vorhersagen des Modells von seinen Eingaben abhängen. Weitere Informationen finden Sie unter Interpretierbarkeit des Modells für maschinelles Lernen mit. AWS

IoT

Siehe Internet der Dinge.

IT information library (ITIL, IT-Informationsbibliothek)

Eine Reihe von bewährten Methoden für die Bereitstellung von IT-Services und die Abstimmung dieser Services auf die Geschäftsanforderungen. ITIL bietet die Grundlage für ITSM.

T service management (ITSM, IT-Servicemanagement)

Aktivitäten im Zusammenhang mit der Gestaltung, Implementierung, Verwaltung und Unterstützung von IT-Services für eine Organisation. Informationen zur Integration von Cloud-Vorgängen mit ITSM-Tools finden Sie im Leitfaden zur Betriebsintegration.

BIS

Siehe IT-Informationsbibliothek.

ITSM

Siehe IT-Servicemanagement.

L

Labelbasierte Zugangskontrolle (LBAC)

Eine Implementierung der Mandatory Access Control (MAC), bei der den Benutzern und den Daten selbst jeweils explizit ein Sicherheitslabelwert zugewiesen wird. Die Schnittmenge zwischen der Benutzersicherheitsbeschriftung und der Datensicherheitsbeschriftung bestimmt, welche Zeilen und Spalten für den Benutzer sichtbar sind.

Landing Zone

Eine landing zone ist eine gut strukturierte AWS Umgebung mit mehreren Konten, die skalierbar und sicher ist. Dies ist ein Ausgangspunkt, von dem aus Ihre Organisationen Workloads und Anwendungen schnell und mit Vertrauen in ihre Sicherheits- und Infrastrukturumgebung starten

und bereitstellen können. Weitere Informationen zu Landing Zones finden Sie unter Einrichtung einer sicheren und skalierbaren AWS -Umgebung mit mehreren Konten..

großes Sprachmodell (LLM)

Ein Deep-Learning-KI-Modell, das anhand einer riesigen Datenmenge vorab trainiert wurde. Ein LLM kann mehrere Aufgaben ausführen, z. B. Fragen beantworten, Dokumente zusammenfassen, Text in andere Sprachen übersetzen und Sätze vervollständigen. Weitere Informationen finden Sie unter Was sind. LLMs

Große Migration

Eine Migration von 300 oder mehr Servern.

SCHWARZ

Siehe Labelbasierte Zugriffskontrolle.

Geringste Berechtigung

Die bewährte Sicherheitsmethode, bei der nur die für die Durchführung einer Aufgabe erforderlichen Mindestberechtigungen erteilt werden. Weitere Informationen finden Sie unter Geringste Berechtigungen anwenden in der IAM-Dokumentation.

Lift and Shift

Siehe 7 Rs.

Little-Endian-System

Ein System, welches das niedrigwertigste Byte zuerst speichert. Siehe auch Endianness.

LLM

Siehe großes Sprachmodell.

Niedrigere Umgebungen

Siehe Umgebung.

M

Machine Learning (ML)

Eine Art künstlicher Intelligenz, die Algorithmen und Techniken zur Mustererkennung und zum Lernen verwendet. ML analysiert aufgezeichnete Daten, wie z. B. Daten aus dem Internet der

Dinge (IoT), und lernt daraus, um ein statistisches Modell auf der Grundlage von Mustern zu erstellen. Weitere Informationen finden Sie unter Machine Learning.

Hauptzweig

Siehe Filiale.

Malware

Software, die entwickelt wurde, um die Computersicherheit oder den Datenschutz zu gefährden. Malware kann Computersysteme stören, vertrauliche Informationen durchsickern lassen oder sich unbefugten Zugriff verschaffen. Beispiele für Malware sind Viren, Würmer, Ransomware, Trojaner, Spyware und Keylogger.

verwaltete Dienste

AWS-Services für die die Infrastrukturebene, das Betriebssystem und die Plattformen AWS betrieben werden, und Sie greifen auf die Endgeräte zu, um Daten zu speichern und abzurufen. Amazon Simple Storage Service (Amazon S3) und Amazon DynamoDB sind Beispiele für Managed Services. Diese werden auch als abstrakte Dienste bezeichnet.

Manufacturing Execution System (MES)

Ein Softwaresystem zur Verfolgung, Überwachung, Dokumentation und Steuerung von Produktionsprozessen, bei denen Rohstoffe in der Fertigung zu fertigen Produkten umgewandelt werden.

MAP

Siehe Migration Acceleration Program.

Mechanismus

Ein vollständiger Prozess, bei dem Sie ein Tool erstellen, die Akzeptanz des Tools vorantreiben und anschließend die Ergebnisse überprüfen, um Anpassungen vorzunehmen. Ein Mechanismus ist ein Zyklus, der sich im Laufe seiner Tätigkeit selbst verstärkt und verbessert. Weitere Informationen finden Sie unter Aufbau von Mechanismen im AWS Well-Architected Framework.

Mitgliedskonto

Alle AWS-Konten außer dem Verwaltungskonto, die Teil einer Organisation in sind. AWS Organizations Ein Konto kann jeweils nur Mitglied einer Organisation sein.

MES

Siehe Manufacturing Execution System.

Message Queuing-Telemetrietransport (MQTT)

Ein leichtes machine-to-machine (M2M) -Kommunikationsprotokoll, das auf dem Publish/Subscribe-Muster für IoT-Geräte mit beschränkten Ressourcen basiert.

Microservice

Ein kleiner, unabhängiger Dienst, der über genau definierte Kanäle kommuniziert APIs und in der Regel kleinen, eigenständigen Teams gehört. Ein Versicherungssystem kann beispielsweise Microservices beinhalten, die Geschäftsfunktionen wie Vertrieb oder Marketing oder Subdomains wie Einkauf, Schadenersatz oder Analytik zugeordnet sind. Zu den Vorteilen von Microservices gehören Agilität, flexible Skalierung, einfache Bereitstellung, wiederverwendbarer Code und Ausfallsicherheit. Weitere Informationen finden Sie unter Integration von Microservices mithilfe serverloser Dienste. AWS

Microservices-Architekturen

Ein Ansatz zur Erstellung einer Anwendung mit unabhängigen Komponenten, die jeden Anwendungsprozess als Microservice ausführen. Diese Microservices kommunizieren mithilfe von Lightweight über eine klar definierte Schnittstelle. APIs Jeder Microservice in dieser Architektur kann aktualisiert, bereitgestellt und skaliert werden, um den Bedarf an bestimmten Funktionen einer Anwendung zu decken. Weitere Informationen finden Sie unter Implementierung von Microservices auf. AWS

Migration Acceleration Program (MAP)

Ein AWS Programm, das Beratung, Unterstützung, Schulungen und Services bietet, um Unternehmen dabei zu unterstützen, eine solide betriebliche Grundlage für die Umstellung auf die Cloud zu schaffen und die anfänglichen Kosten von Migrationen auszugleichen. MAP umfasst eine Migrationsmethode für die methodische Durchführung von Legacy-Migrationen sowie eine Reihe von Tools zur Automatisierung und Beschleunigung gängiger Migrationsszenarien.

Migration in großem Maßstab

Der Prozess, bei dem der Großteil des Anwendungsportfolios in Wellen in die Cloud verlagert wird, wobei in jeder Welle mehr Anwendungen schneller migriert werden. In dieser Phase werden die bewährten Verfahren und Erkenntnisse aus den früheren Phasen zur Implementierung einer Migrationsfabrik von Teams, Tools und Prozessen zur Optimierung der Migration von Workloads durch Automatisierung und agile Bereitstellung verwendet. Dies ist die dritte Phase der AWS - Migrationsstrategie.

Migrationsfabrik

Funktionsübergreifende Teams, die die Migration von Workloads durch automatisierte, agile Ansätze optimieren. Zu den Teams in der Migrationsabteilung gehören in der Regel Betriebsabläufe, Geschäftsanalysten und Eigentümer, Migrationsingenieure, Entwickler und DevOps Experten, die in Sprints arbeiten. Zwischen 20 und 50 Prozent eines Unternehmensanwendungsportfolios bestehen aus sich wiederholenden Mustern, die durch einen Fabrik-Ansatz optimiert werden können. Weitere Informationen finden Sie in Diskussion über Migrationsfabriken und den Leitfaden zur Cloud-Migration-Fabrik in diesem Inhaltssatz.

Migrationsmetadaten

Die Informationen über die Anwendung und den Server, die für den Abschluss der Migration benötigt werden. Für jedes Migrationsmuster ist ein anderer Satz von Migrationsmetadaten erforderlich. Beispiele für Migrationsmetadaten sind das Zielsubnetz, die Sicherheitsgruppe und AWS das Konto.

Migrationsmuster

Eine wiederholbare Migrationsaufgabe, in der die Migrationsstrategie, das Migrationsziel und die verwendete Migrationsanwendung oder der verwendete Migrationsservice detailliert beschrieben werden. Beispiel: Rehost-Migration zu Amazon EC2 mit AWS Application Migration Service.

Migration Portfolio Assessment (MPA)

Ein Online-Tool, das Informationen zur Validierung des Geschäftsszenarios für die Migration auf das bereitstellt. AWS Cloud MPA bietet eine detaillierte Portfoliobewertung (richtige Servergröße, Preisgestaltung, Gesamtbetriebskostenanalyse, Migrationskostenanalyse) sowie Migrationsplanung (Anwendungsdatenanalyse und Datenerfassung, Anwendungsgruppierung, Migrationspriorisierung und Wellenplanung). Das MPA-Tool (Anmeldung erforderlich) steht allen AWS Beratern und APN-Partnerberatern kostenlos zur Verfügung.

Migration Readiness Assessment (MRA)

Der Prozess, bei dem mithilfe des AWS CAF Erkenntnisse über den Cloud-Bereitschaftsstatus eines Unternehmens gewonnen, Stärken und Schwächen identifiziert und ein Aktionsplan zur Schließung festgestellter Lücken erstellt wird. Weitere Informationen finden Sie im Benutzerhandbuch für Migration Readiness. MRA ist die erste Phase der AWS - Migrationsstrategie.

Migrationsstrategie

Der Ansatz, der verwendet wurde, um einen Workload auf den AWS Cloud zu migrieren. Weitere Informationen finden Sie im Eintrag 7 Rs in diesem Glossar und unter Mobilisieren Sie Ihr Unternehmen, um umfangreiche Migrationen zu beschleunigen.

ML

Siehe maschinelles Lernen.

Modernisierung

Umwandlung einer veralteten (veralteten oder monolithischen) Anwendung und ihrer Infrastruktur in ein agiles, elastisches und hochverfügbares System in der Cloud, um Kosten zu senken, die Effizienz zu steigern und Innovationen zu nutzen. Weitere Informationen finden Sie unter Strategie zur Modernisierung von Anwendungen in der AWS Cloud.

Bewertung der Modernisierungsfähigkeit

Eine Bewertung, anhand derer festgestellt werden kann, ob die Anwendungen einer Organisation für die Modernisierung bereit sind, Vorteile, Risiken und Abhängigkeiten identifiziert und ermittelt wird, wie gut die Organisation den zukünftigen Status dieser Anwendungen unterstützen kann. Das Ergebnis der Bewertung ist eine Vorlage der Zielarchitektur, eine Roadmap, in der die Entwicklungsphasen und Meilensteine des Modernisierungsprozesses detailliert beschrieben werden, sowie ein Aktionsplan zur Behebung festgestellter Lücken. Weitere Informationen finden Sie unter Evaluierung der Modernisierungsbereitschaft von Anwendungen in der AWS Cloud.

Monolithische Anwendungen (Monolithen)

Anwendungen, die als ein einziger Service mit eng gekoppelten Prozessen ausgeführt werden. Monolithische Anwendungen haben verschiedene Nachteile. Wenn ein Anwendungs-Feature stark nachgefragt wird, muss die gesamte Architektur skaliert werden. Das Hinzufügen oder Verbessern der Feature einer monolithischen Anwendung wird ebenfalls komplexer, wenn die Codebasis wächst. Um diese Probleme zu beheben, können Sie eine Microservices-Architektur verwenden. Weitere Informationen finden Sie unter Zerlegen von Monolithen in Microservices.

MPA

Siehe Bewertung des Migrationsportfolios.

MQTT

Siehe Message Queuing-Telemetrietransport.

Mehrklassen-Klassifizierung

Ein Prozess, der dabei hilft, Vorhersagen für mehrere Klassen zu generieren (wobei eines von mehr als zwei Ergebnissen vorhergesagt wird). Ein ML-Modell könnte beispielsweise fragen: „Ist dieses Produkt ein Buch, ein Auto oder ein Telefon?“ oder „Welche Kategorie von Produkten ist für diesen Kunden am interessantesten?“

veränderbare Infrastruktur

Ein Modell, das die bestehende Infrastruktur für Produktionsworkloads aktualisiert und modifiziert. Für eine verbesserte Konsistenz, Zuverlässigkeit und Vorhersagbarkeit empfiehlt das AWS Well-Architected Framework die Verwendung einer unveränderlichen Infrastruktur als bewährte Methode.

O

OAC

Siehe Origin Access Control.

EICHE

Siehe Zugriffsidentität von Origin.

COM

Siehe organisatorisches Change-Management.

Offline-Migration

Eine Migrationsmethode, bei der der Quell-Workload während des Migrationsprozesses heruntergefahren wird. Diese Methode ist mit längeren Ausfallzeiten verbunden und wird in der Regel für kleine, unkritische Workloads verwendet.

OI

Siehe Betriebsintegration.

OLA

Siehe Vereinbarung auf operativer Ebene.

Online-Migration

Eine Migrationsmethode, bei der der Quell-Workload auf das Zielsystem kopiert wird, ohne offline genommen zu werden. Anwendungen, die mit dem Workload verbunden sind, können während

der Migration weiterhin funktionieren. Diese Methode beinhaltet keine bis minimale Ausfallzeit und wird in der Regel für kritische Produktionsworkloads verwendet.

OPC-UA

Siehe Open Process Communications — Unified Architecture.

Offene Prozesskommunikation — Einheitliche Architektur (OPC-UA)

Ein machine-to-machine (M2M) -Kommunikationsprotokoll für die industrielle Automatisierung. OPC-UA bietet einen Interoperabilitätsstandard mit Datenverschlüsselungs-, Authentifizierungs- und Autorisierungsschemata.

Vereinbarung auf Betriebsebene (OLA)

Eine Vereinbarung, in der klargestellt wird, welche funktionalen IT-Gruppen sich gegenseitig versprechen zu liefern, um ein Service Level Agreement (SLA) zu unterstützen.

Überprüfung der Betriebsbereitschaft (ORR)

Eine Checkliste mit Fragen und zugehörigen bewährten Methoden, die Ihnen helfen, Vorfälle und mögliche Ausfälle zu verstehen, zu bewerten, zu verhindern oder deren Umfang zu reduzieren. Weitere Informationen finden Sie unter Operational Readiness Reviews (ORR) im AWS Well-Architected Framework.

Betriebstechnologie (OT)

Hardware- und Softwaresysteme, die mit der physischen Umgebung zusammenarbeiten, um industrielle Abläufe, Ausrüstung und Infrastruktur zu steuern. In der Fertigung ist die Integration von OT- und Informationstechnologie (IT) -Systemen ein zentraler Schwerpunkt der Industrie 4.0-Transformationen.

Betriebsintegration (OI)

Der Prozess der Modernisierung von Abläufen in der Cloud, der Bereitschaftsplanung, Automatisierung und Integration umfasst. Weitere Informationen finden Sie im Leitfaden zur Betriebsintegration.

Organisationspfad

Ein Pfad, der von erstellt wird und in AWS CloudTrail dem alle Ereignisse für alle AWS-Konten in einer Organisation protokolliert werden. AWS Organizations Diese Spur wird in jedem AWS-Konto , der Teil der Organisation ist, erstellt und verfolgt die Aktivität in jedem Konto. Weitere Informationen finden Sie in der CloudTrail Dokumentation unter Einen Trail für eine Organisation erstellen.

Organisatorisches Veränderungsmanagement (OCM)

Ein Framework für das Management wichtiger, disruptiver Geschäftstransformationen aus Sicht der Mitarbeiter, der Kultur und der Führung. OCM hilft Organisationen dabei, sich auf neue Systeme und Strategien vorzubereiten und auf diese umzustellen, indem es die Akzeptanz von Veränderungen beschleunigt, Übergangsprobleme angeht und kulturelle und organisatorische Veränderungen vorantreibt. In der AWS Migrationsstrategie wird dieses Framework aufgrund der Geschwindigkeit des Wandels, der bei Projekten zur Cloud-Einführung erforderlich ist, als Mitarbeiterbeschleunigung bezeichnet. Weitere Informationen finden Sie im OCM-Handbuch.

Ursprungszugriffskontrolle (OAC)

In CloudFront, eine erweiterte Option zur Zugriffsbeschränkung, um Ihre Amazon Simple Storage Service (Amazon S3) -Inhalte zu sichern. OAC unterstützt alle S3-Buckets insgesamt AWS-Regionen, serverseitige Verschlüsselung mit AWS KMS (SSE-KMS) sowie dynamische PUT und `DELETE` Anfragen an den S3-Bucket.

Ursprungszugriffsidentität (OAI)

In CloudFront, eine Option zur Zugriffsbeschränkung, um Ihre Amazon S3 S3-Inhalte zu sichern. Wenn Sie OAI verwenden, CloudFront erstellt es einen Principal, mit dem sich Amazon S3 authentifizieren kann. Authentifizierte Principals können nur über eine bestimmte Distribution auf Inhalte in einem S3-Bucket zugreifen. CloudFront Siehe auch OAC, das eine detailliertere und verbesserte Zugriffskontrolle bietet.

ORR

Weitere Informationen finden Sie unter Überprüfung der Betriebsbereitschaft.

NICHT

Siehe Betriebstechnologie.

Ausgehende (egress) VPC

In einer Architektur AWS mit mehreren Konten eine VPC, die Netzwerkverbindungen verarbeitet, die von einer Anwendung aus initiiert werden. Die AWS Security Reference Architecture empfiehlt die Einrichtung Ihres Netzwerkkontos mit eingehendem und ausgehendem Datenverkehr sowie Inspektion, VPCs um die bidirektionale Schnittstelle zwischen Ihrer Anwendung und dem Internet im weiteren Sinne zu schützen.

P

Berechtigungsgrenze

Eine IAM-Verwaltungsrichtlinie, die den IAM-Prinzipalen zugeordnet ist, um die maximalen Berechtigungen festzulegen, die der Benutzer oder die Rolle haben kann. Weitere Informationen finden Sie unter Berechtigungsgrenzen für IAM-Entitys in der IAM-Dokumentation.

persönlich identifizierbare Informationen (PII)

Informationen, die, wenn sie direkt betrachtet oder mit anderen verwandten Daten kombiniert werden, verwendet werden können, um vernünftige Rückschlüsse auf die Identität einer Person zu ziehen. Beispiele für personenbezogene Daten sind Namen, Adressen und Kontaktinformationen.

Personenbezogene Daten

Siehe persönlich identifizierbare Informationen.

Playbook

Eine Reihe vordefinierter Schritte, die die mit Migrationen verbundenen Aufgaben erfassen, z. B. die Bereitstellung zentraler Betriebsfunktionen in der Cloud. Ein Playbook kann die Form von Skripten, automatisierten Runbooks oder einer Zusammenfassung der Prozesse oder Schritte annehmen, die für den Betrieb Ihrer modernisierten Umgebung erforderlich sind.

PLC

Siehe programmierbare Logiksteuerung.

PLM

Siehe Produktlebenszyklusmanagement.

policy

Ein Objekt, das Berechtigungen definieren (siehe identitätsbasierte Richtlinie), Zugriffsbedingungen spezifizieren (siehe ressourcenbasierte Richtlinie) oder die maximalen Berechtigungen für alle Konten in einer Organisation definieren kann AWS Organizations (siehe Dienststeuerungsrichtlinie).

Polyglotte Beharrlichkeit

Unabhängige Auswahl der Datenspeichertechnologie eines Microservices auf der Grundlage von Datenzugriffsmustern und anderen Anforderungen. Wenn Ihre Microservices über dieselbe Datenspeichertechnologie verfügen, kann dies zu Implementierungsproblemen oder zu

Leistungseinbußen führen. Microservices lassen sich leichter implementieren und erzielen eine bessere Leistung und Skalierbarkeit, wenn sie den Datenspeicher verwenden, der ihren Anforderungen am besten entspricht.

Portfoliobewertung

Ein Prozess, bei dem das Anwendungsportfolio ermittelt, analysiert und priorisiert wird, um die Migration zu planen. Weitere Informationen finden Sie in Bewerten der Migrationsbereitschaft.

predicate

Eine Abfragebedingung, die `true` oder zurückgibt`false`, was üblicherweise in einer Klausel vorkommt. `WHERE`

Prädikat Pushdown

Eine Technik zur Optimierung von Datenbankabfragen, bei der die Daten in der Abfrage vor der Übertragung gefiltert werden. Dadurch wird die Datenmenge reduziert, die aus der relationalen Datenbank abgerufen und verarbeitet werden muss, und die Abfrageleistung wird verbessert.

Präventive Kontrolle

Eine Sicherheitskontrolle, die verhindern soll, dass ein Ereignis eintritt. Diese Kontrollen stellen eine erste Verteidigungslinie dar, um unbefugten Zugriff oder unerwünschte Änderungen an Ihrem Netzwerk zu verhindern. Weitere Informationen finden Sie unter Präventive Kontrolle in Implementierung von Sicherheitskontrollen in AWS.

Prinzipal

Eine Entität AWS , die Aktionen ausführen und auf Ressourcen zugreifen kann. Diese Entität ist in der Regel ein Root-Benutzer für eine AWS-Konto, eine IAM-Rolle oder einen Benutzer. Weitere Informationen finden Sie unter Prinzipal in Rollenbegriffe und -konzepte in der IAM-Dokumentation.

Datenschutz von Natur aus

Ein systemtechnischer Ansatz, der den Datenschutz während des gesamten Entwicklungsprozesses berücksichtigt.

Privat gehostete Zonen

Ein Container, der Informationen darüber enthält, wie Amazon Route 53 auf DNS-Abfragen für eine Domain und deren Subdomains innerhalb einer oder mehrerer VPCs Domains antworten soll. Weitere Informationen finden Sie unter Arbeiten mit privat gehosteten Zonen in der Route-53-Dokumentation.

proaktive Steuerung

Eine Sicherheitskontrolle, die den Einsatz nicht richtlinienkonformer Ressourcen verhindern soll. Diese Steuerelemente scannen Ressourcen, bevor sie bereitgestellt werden. Wenn die Ressource nicht der Kontrolle entspricht, wird sie nicht bereitgestellt. Weitere Informationen finden Sie im Referenzhandbuch zu Kontrollen in der AWS Control Tower Dokumentation und unter Proaktive Kontrollen unter Implementierung von Sicherheitskontrollen am AWS.

Produktlebenszyklusmanagement (PLM)

Das Management von Daten und Prozessen für ein Produkt während seines gesamten Lebenszyklus, vom Design, der Entwicklung und Markteinführung über Wachstum und Reife bis hin zur Markteinführung und Markteinführung.

Produktionsumgebung

Siehe Umgebung.

Speicherprogrammierbare Steuerung (SPS)

In der Fertigung ein äußerst zuverlässiger, anpassungsfähiger Computer, der Maschinen überwacht und Fertigungsprozesse automatisiert.

schnelle Verkettung

Verwendung der Ausgabe einer LLM-Eingabeaufforderung als Eingabe für die nächste Aufforderung, um bessere Antworten zu generieren. Diese Technik wird verwendet, um eine komplexe Aufgabe in Unteraufgaben zu unterteilen oder um eine vorläufige Antwort iterativ zu verfeinern oder zu erweitern. Sie trägt dazu bei, die Genauigkeit und Relevanz der Antworten eines Modells zu verbessern und ermöglicht detailliertere, personalisierte Ergebnisse.

Pseudonymisierung

Der Prozess, bei dem persönliche Identifikatoren in einem Datensatz durch Platzhalterwerte ersetzt werden. Pseudonymisierung kann zum Schutz der Privatsphäre beitragen. Pseudonymisierte Daten gelten weiterhin als personenbezogene Daten.

publish/subscribe (pub/sub)

Ein Muster, das asynchrone Kommunikation zwischen Microservices ermöglicht, um die Skalierbarkeit und Reaktionsfähigkeit zu verbessern. In einem auf Microservices basierenden MES kann ein Microservice beispielsweise Ereignismeldungen in einem Kanal veröffentlichen, den andere Microservices abonnieren können. Das System kann neue Microservices hinzufügen, ohne den Veröffentlichungsservice zu ändern.

Q

Abfrageplan

Eine Reihe von Schritten, wie Anweisungen, die für den Zugriff auf die Daten in einem relationalen SQL-Datenbanksystem verwendet werden.

Abfrageplanregression

Wenn ein Datenbankserviceoptimierer einen weniger optimalen Plan wählt als vor einer bestimmten Änderung der Datenbankumgebung. Dies kann durch Änderungen an Statistiken, Beschränkungen, Umgebungseinstellungen, Abfrageparameter-Bindungen und Aktualisierungen der Datenbank-Engine verursacht werden.

R

RACI-Matrix

Siehe verantwortlich, rechenschaftspflichtig, konsultiert, informiert (RACI).

RAG

Siehe Erweiterte Generierung beim Abrufen.

Ransomware

Eine bösartige Software, die entwickelt wurde, um den Zugriff auf ein Computersystem oder Daten zu blockieren, bis eine Zahlung erfolgt ist.

RASCI-Matrix

Siehe verantwortlich, rechenschaftspflichtig, konsultiert, informiert (RACI).

RCAC

Siehe Zugriffskontrolle für Zeilen und Spalten.

Read Replica

Eine Kopie einer Datenbank, die nur für Lesezwecke verwendet wird. Sie können Abfragen an das Lesereplikat weiterleiten, um die Belastung auf Ihrer Primärdatenbank zu reduzieren.

neu strukturieren

Siehe 7 Rs.

Recovery Point Objective (RPO)

Die maximal zulässige Zeitspanne seit dem letzten Datenwiederherstellungspunkt. Damit wird festgelegt, was als akzeptabler Datenverlust zwischen dem letzten Wiederherstellungspunkt und der Serviceunterbrechung gilt.

Wiederherstellungszeitziel (RTO)

Die maximal zulässige Verzögerung zwischen der Betriebsunterbrechung und der Wiederherstellung des Dienstes.

Refaktorierung

Siehe 7 Rs.

Region

Eine Sammlung von AWS Ressourcen in einem geografischen Gebiet. Jeder AWS-Region ist isoliert und unabhängig von den anderen, um Fehlertoleranz, Stabilität und Belastbarkeit zu gewährleisten. Weitere Informationen finden Sie unter Geben Sie an, was AWS-Regionen Ihr Konto verwenden kann.

Regression

Eine ML-Technik, die einen numerischen Wert vorhersagt. Zum Beispiel, um das Problem „Zu welchem Preis wird dieses Haus verkauft werden?“ zu lösen Ein ML-Modell könnte ein lineares Regressionsmodell verwenden, um den Verkaufspreis eines Hauses auf der Grundlage bekannter Fakten über das Haus (z. B. die Quadratmeterzahl) vorherzusagen.

rehosten

Siehe 7 Rs.

Veröffentlichung

In einem Bereitstellungsprozess der Akt der Förderung von Änderungen an einer Produktionsumgebung.

umziehen

Siehe 7 Rs.

neue Plattform

Siehe 7 Rs.

Rückkauf

Siehe 7 Rs.

Ausfallsicherheit

Die Fähigkeit einer Anwendung, Störungen zu widerstehen oder sich von ihnen zu erholen. Hochverfügbarkeit und Notfallwiederherstellung sind häufig Überlegungen bei der Planung der Ausfallsicherheit in der. AWS Cloud Weitere Informationen finden Sie unter AWS Cloud Resilienz.

Ressourcenbasierte Richtlinie

Eine mit einer Ressource verknüpfte Richtlinie, z. B. ein Amazon-S3-Bucket, ein Endpunkt oder ein Verschlüsselungsschlüssel. Diese Art von Richtlinie legt fest, welchen Prinzipalen der Zugriff gewährt wird, welche Aktionen unterstützt werden und welche anderen Bedingungen erfüllt sein müssen.

RACI-Matrix (verantwortlich, rechenschaftspflichtig, konsultiert, informiert)

Eine Matrix, die die Rollen und Verantwortlichkeiten aller an Migrationsaktivitäten und Cloud-Operationen beteiligten Parteien definiert. Der Matrixname leitet sich von den in der Matrix definierten Zuständigkeitstypen ab: verantwortlich (R), rechenschaftspflichtig (A), konsultiert (C) und informiert (I). Der Unterstützungstyp (S) ist optional. Wenn Sie Unterstützung einbeziehen, wird die Matrix als RASCI-Matrix bezeichnet, und wenn Sie sie ausschließen, wird sie als RACI-Matrix bezeichnet.

Reaktive Kontrolle

Eine Sicherheitskontrolle, die darauf ausgelegt ist, die Behebung unerwünschter Ereignisse oder Abweichungen von Ihren Sicherheitsstandards voranzutreiben. Weitere Informationen finden Sie unter Reaktive Kontrolle in Implementieren von Sicherheitskontrollen in AWS.

Beibehaltung

Siehe 7 Rs.

zurückziehen

Siehe 7 Rs.

Retrieval Augmented Generation (RAG)

Eine generative KI-Technologie, bei der ein LLM auf eine maßgebliche Datenquelle verweist, die sich außerhalb seiner Trainingsdatenquellen befindet, bevor eine Antwort generiert wird. Ein RAG-Modell könnte beispielsweise eine semantische Suche in der Wissensdatenbank oder in

benutzerdefinierten Daten einer Organisation durchführen. Weitere Informationen finden Sie unter Was ist RAG.

Drehung

Der Vorgang, bei dem ein Geheimnis regelmäßig aktualisiert wird, um es einem Angreifer zu erschweren, auf die Anmeldeinformationen zuzugreifen.

Zugriffskontrolle für Zeilen und Spalten (RCAC)

Die Verwendung einfacher, flexibler SQL-Ausdrücke mit definierten Zugriffsregeln. RCAC besteht aus Zeilenberechtigungen und Spaltenmasken.

RPO

Siehe Recovery Point Objective.

RTO

Siehe Ziel für die Erholungszeit.

Runbook

Eine Reihe manueller oder automatisierter Verfahren, die zur Ausführung einer bestimmten Aufgabe erforderlich sind. Diese sind in der Regel darauf ausgelegt, sich wiederholende Operationen oder Verfahren mit hohen Fehlerquoten zu rationalisieren.

S

SAML 2.0

Ein offener Standard, den viele Identitätsanbieter (IdPs) verwenden. Diese Funktion ermöglicht föderiertes Single Sign-On (SSO), sodass sich Benutzer bei den API-Vorgängen anmelden AWS-Managementkonsole oder die AWS API-Operationen aufrufen können, ohne dass Sie einen Benutzer in IAM für alle in Ihrer Organisation erstellen müssen. Weitere Informationen zum SAML-2.0.-basierten Verbund finden Sie unter Über den SAML-2.0-basierten Verbund in der IAM-Dokumentation.

SCADA

Siehe Aufsichtskontrolle und Datenerfassung.

SCP

Siehe Richtlinie zur Dienstkontrolle.

Secret

Interne AWS Secrets Manager, vertrauliche oder eingeschränkte Informationen, wie z. B. ein Passwort oder Benutzeranmeldeinformationen, die Sie in verschlüsselter Form speichern. Es besteht aus dem geheimen Wert und seinen Metadaten. Der geheime Wert kann binär, eine einzelne Zeichenfolge oder mehrere Zeichenketten sein. Weitere Informationen finden Sie unter Was ist in einem Secrets Manager Manager-Geheimnis? in der Secrets Manager Manager-Dokumentation.

Sicherheit durch Design

Ein systemtechnischer Ansatz, der die Sicherheit während des gesamten Entwicklungsprozesses berücksichtigt.

Sicherheitskontrolle

Ein technischer oder administrativer Integritätsschutz, der die Fähigkeit eines Bedrohungsakteurs, eine Schwachstelle auszunutzen, verhindert, erkennt oder einschränkt. Es gibt vier Haupttypen von Sicherheitskontrollen: präventiv, detektiv, reaktionsschnell und proaktiv.

Härtung der Sicherheit

Der Prozess, bei dem die Angriffsfläche reduziert wird, um sie widerstandsfähiger gegen Angriffe zu machen. Dies kann Aktionen wie das Entfernen von Ressourcen, die nicht mehr benötigt werden, die Implementierung der bewährten Sicherheitsmethode der Gewährung geringster Berechtigungen oder die Deaktivierung unnötiger Feature in Konfigurationsdateien umfassen.

System zur Verwaltung von Sicherheitsinformationen und Ereignissen (security information and event management – SIEM)

Tools und Services, die Systeme für das Sicherheitsinformationsmanagement (SIM) und das Management von Sicherheitsereignissen (SEM) kombinieren. Ein SIEM-System sammelt, überwacht und analysiert Daten von Servern, Netzwerken, Geräten und anderen Quellen, um Bedrohungen und Sicherheitsverletzungen zu erkennen und Warnmeldungen zu generieren.

Automatisierung von Sicherheitsreaktionen

Eine vordefinierte und programmierte Aktion, die darauf ausgelegt ist, automatisch auf ein Sicherheitsereignis zu reagieren oder es zu beheben. Diese Automatisierungen dienen als detektive oder reaktionsschnelle Sicherheitskontrollen, die Sie bei der Implementierung bewährter AWS Sicherheitsmethoden unterstützen. Beispiele für automatisierte Antwortaktionen sind das Ändern einer VPC-Sicherheitsgruppe, das Patchen einer Amazon EC2 EC2-Instance oder das Rotieren von Anmeldeinformationen.

Serverseitige Verschlüsselung

Verschlüsselung von Daten am Zielort durch denjenigen AWS-Service , der sie empfängt.

Service-Kontrollrichtlinie (SCP)

Eine Richtlinie, die eine zentrale Steuerung der Berechtigungen für alle Konten in einer Organisation in ermöglicht AWS Organizations. SCPs Definieren Sie Leitplanken oder legen Sie Grenzwerte für Aktionen fest, die ein Administrator an Benutzer oder Rollen delegieren kann. Sie können sie SCPs als Zulassungs- oder Ablehnungslisten verwenden, um festzulegen, welche Dienste oder Aktionen zulässig oder verboten sind. Weitere Informationen finden Sie in der AWS Organizations Dokumentation unter Richtlinien zur Dienststeuerung.

Service-Endpunkt

Die URL des Einstiegspunkts für einen AWS-Service. Sie können den Endpunkt verwenden, um programmgesteuert eine Verbindung zum Zielservice herzustellen. Weitere Informationen finden Sie unter AWS-Service -Endpunkte in der Allgemeine AWS-Referenz.

Service Level Agreement (SLA)

Eine Vereinbarung, in der klargestellt wird, was ein IT-Team seinen Kunden zu bieten verspricht, z. B. in Bezug auf Verfügbarkeit und Leistung der Services.

Service-Level-Indikator (SLI)

Eine Messung eines Leistungsaspekts eines Dienstes, z. B. seiner Fehlerrate, Verfügbarkeit oder Durchsatz.

Service-Level-Ziel (SLO)

Eine Zielkennzahl, die den Zustand eines Dienstes darstellt, gemessen anhand eines Service-Level-Indikators.

Modell der geteilten Verantwortung

Ein Modell, das die Verantwortung beschreibt, mit der Sie gemeinsam AWS für Cloud-Sicherheit und Compliance verantwortlich sind. AWS ist für die Sicherheit der Cloud verantwortlich, während Sie für die Sicherheit in der Cloud verantwortlich sind. Weitere Informationen finden Sie unter Modell der geteilten Verantwortung.

SIEM

Siehe Sicherheitsinformations- und Event-Management-System.

Single Point of Failure (SPOF)

Ein Fehler in einer einzelnen, kritischen Komponente einer Anwendung, der das System stören kann.

SLA

Siehe Service Level Agreement.

SLI

Siehe Service-Level-Indikator.

ALSO

Siehe Service-Level-Ziel.

split-and-seed Modell

Ein Muster für die Skalierung und Beschleunigung von Modernisierungsprojekten. Sobald neue Features und Produktversionen definiert werden, teilt sich das Kernteam auf, um neue Produktteams zu bilden. Dies trägt zur Skalierung der Fähigkeiten und Services Ihrer Organisation bei, verbessert die Produktivität der Entwickler und unterstützt schnelle Innovationen. Weitere Informationen finden Sie unter Schrittweiser Ansatz zur Modernisierung von Anwendungen in der AWS Cloud.

SPOTTEN

Siehe Single Point of Failure.

Sternschema

Eine Datenbank-Organisationsstruktur, die eine große Faktentabelle zum Speichern von Transaktions- oder Messdaten und eine oder mehrere kleinere dimensionale Tabellen zum Speichern von Datenattributen verwendet. Diese Struktur ist für die Verwendung in einem Data Warehouse oder für Business Intelligence-Zwecke konzipiert.

Strangler-Fig-Muster

Ein Ansatz zur Modernisierung monolithischer Systeme, bei dem die Systemfunktionen schrittweise umgeschrieben und ersetzt werden, bis das Legacy-System außer Betrieb genommen werden kann. Dieses Muster verwendet die Analogie einer Feigenrebe, die zu einem etablierten Baum heranwächst und schließlich ihren Wirt überwindet und ersetzt. Das Muster wurde eingeführt von Martin Fowler als Möglichkeit, Risiken beim Umschreiben monolithischer Systeme zu managen. Ein Beispiel für die Anwendung dieses Musters finden Sie

unter Schrittweises Modernisieren älterer Microsoft ASP.NET (ASMX)-Webservices mithilfe von Containern und Amazon API Gateway.

Subnetz

Ein Bereich von IP-Adressen in Ihrer VPC. Ein Subnetz muss sich in einer einzigen Availability Zone befinden.

Aufsichtskontrolle und Datenerfassung (SCADA)

In der Fertigung ein System, das Hardware und Software zur Überwachung von Sachanlagen und Produktionsabläufen verwendet.

Symmetrische Verschlüsselung

Ein Verschlüsselungsalgorithmus, der denselben Schlüssel zum Verschlüsseln und Entschlüsseln der Daten verwendet.

synthetisches Testen

Testen eines Systems auf eine Weise, die Benutzerinteraktionen simuliert, um potenzielle Probleme zu erkennen oder die Leistung zu überwachen. Sie können Amazon CloudWatch Synthetics verwenden, um diese Tests zu erstellen.

Systemaufforderung

Eine Technik, mit der einem LLM Kontext, Anweisungen oder Richtlinien zur Verfügung gestellt werden, um sein Verhalten zu steuern. Systemaufforderungen helfen dabei, den Kontext festzulegen und Regeln für Interaktionen mit Benutzern festzulegen.

T

tags

Schlüssel-Wert-Paare, die als Metadaten für die Organisation Ihrer Ressourcen dienen. AWS Mit Tags können Sie Ressourcen verwalten, identifizieren, organisieren, suchen und filtern. Weitere Informationen finden Sie unter Markieren Ihrer AWS -Ressourcen.

Zielvariable

Der Wert, den Sie in überwachtem ML vorhersagen möchten. Dies wird auch als Ergebnisvariable bezeichnet. In einer Fertigungsumgebung könnte die Zielvariable beispielsweise ein Produktfehler sein.

Aufgabenliste

Ein Tool, das verwendet wird, um den Fortschritt anhand eines Runbooks zu verfolgen. Eine Aufgabenliste enthält eine Übersicht über das Runbook und eine Liste mit allgemeinen Aufgaben, die erledigt werden müssen. Für jede allgemeine Aufgabe werden der geschätzte Zeitaufwand, der Eigentümer und der Fortschritt angegeben.

Testumgebungen

Siehe Umgebung.

Training

Daten für Ihr ML-Modell bereitstellen, aus denen es lernen kann. Die Trainingsdaten müssen die richtige Antwort enthalten. Der Lernalgorithmus findet Muster in den Trainingsdaten, die die Attribute der Input-Daten dem Ziel (die Antwort, die Sie voraussagen möchten) zuordnen. Es gibt ein ML-Modell aus, das diese Muster erfasst. Sie können dann das ML-Modell verwenden, um Voraussagen für neue Daten zu erhalten, bei denen Sie das Ziel nicht kennen.

Transit-Gateway

Ein Netzwerk-Transit-Hub, über den Sie Ihre Netzwerke VPCs und Ihre lokalen Netzwerke miteinander verbinden können. Weitere Informationen finden Sie in der Dokumentation unter Was ist ein Transit-Gateway. AWS Transit Gateway

Stammbasierter Workflow

Ein Ansatz, bei dem Entwickler Feature lokal in einem Feature-Zweig erstellen und testen und diese Änderungen dann im Hauptzweig zusammenführen. Der Hauptzweig wird dann sequentiell für die Entwicklungs-, Vorproduktions- und Produktionsumgebungen erstellt.

Vertrauenswürdiger Zugriff

Gewährung von Berechtigungen für einen Dienst, den Sie angeben, um Aufgaben in Ihrer Organisation AWS Organizations und in deren Konten in Ihrem Namen auszuführen. Der vertrauenswürdige Service erstellt in jedem Konto eine mit dem Service verknüpfte Rolle, wenn diese Rolle benötigt wird, um Verwaltungsaufgaben für Sie auszuführen. Weitere Informationen finden Sie in der AWS Organizations Dokumentation unter Verwendung AWS Organizations mit anderen AWS Diensten.

Optimieren

Aspekte Ihres Trainingsprozesses ändern, um die Genauigkeit des ML-Modells zu verbessern. Sie können das ML-Modell z. B. trainieren, indem Sie einen Beschriftungssatz generieren,

Beschriftungen hinzufügen und diese Schritte dann mehrmals unter verschiedenen Einstellungen wiederholen, um das Modell zu optimieren.

Zwei-Pizzen-Team

Ein kleines DevOps Team, das Sie mit zwei Pizzen ernähren können. Eine Teamgröße von zwei Pizzen gewährleistet die bestmögliche Gelegenheit zur Zusammenarbeit bei der Softwareentwicklung.

U

Unsicherheit

Ein Konzept, das sich auf ungenaue, unvollständige oder unbekannte Informationen bezieht, die die Zuverlässigkeit von prädiktiven ML-Modellen untergraben können. Es gibt zwei Arten von Unsicherheit: Epistemische Unsicherheit wird durch begrenzte, unvollständige Daten verursacht, wohingegen aleatorische Unsicherheit durch Rauschen und Randomisierung verursacht wird, die in den Daten liegt. Weitere Informationen finden Sie im Leitfaden Quantifizieren der Unsicherheit in Deep-Learning-Systemen.

undifferenzierte Aufgaben

Diese Arbeit wird auch als Schwerstarbeit bezeichnet. Dabei handelt es sich um Arbeiten, die zwar für die Erstellung und den Betrieb einer Anwendung erforderlich sind, aber dem Endbenutzer keinen direkten Mehrwert bieten oder keinen Wettbewerbsvorteil bieten. Beispiele für undifferenzierte Aufgaben sind Beschaffung, Wartung und Kapazitätsplanung.

höhere Umgebungen

Siehe Umgebung.

V

Vacuuming

Ein Vorgang zur Datenbankwartung, bei dem die Datenbank nach inkrementellen Aktualisierungen bereinigt wird, um Speicherplatz zurückzugewinnen und die Leistung zu verbessern.

Versionskontrolle

Prozesse und Tools zur Nachverfolgung von Änderungen, z. B. Änderungen am Quellcode in einem Repository.

VPC-Peering

Eine Verbindung zwischen zwei VPCs , die es Ihnen ermöglicht, den Verkehr mithilfe privater IP-Adressen weiterzuleiten. Weitere Informationen finden Sie unter Was ist VPC-Peering? in der Amazon-VPC-Dokumentation.

Schwachstelle

Ein Software- oder Hardwarefehler, der die Sicherheit des Systems beeinträchtigt.

W

Warmer Cache

Ein Puffer-Cache, der aktuelle, relevante Daten enthält, auf die häufig zugegriffen wird. Die Datenbank-Instance kann aus dem Puffer-Cache lesen, was schneller ist als das Lesen aus dem Hauptspeicher oder von der Festplatte.

warme Daten

Daten, auf die selten zugegriffen wird. Bei der Abfrage dieser Art von Daten sind mäßig langsame Abfragen in der Regel akzeptabel.

Fensterfunktion

Eine SQL-Funktion, die eine Berechnung für eine Gruppe von Zeilen durchführt, die sich in irgendeiner Weise auf den aktuellen Datensatz beziehen. Fensterfunktionen sind nützlich für die Verarbeitung von Aufgaben wie die Berechnung eines gleitenden Durchschnitts oder für den Zugriff auf den Wert von Zeilen auf der Grundlage der relativen Position der aktuellen Zeile.

Workload

Ein Workload ist eine Sammlung von Ressourcen und Code, die einen Unternehmenswert bietet, wie z. B. eine kundenorientierte Anwendung oder ein Backend-Prozess.

Workstream

Funktionsgruppen in einem Migrationsprojekt, die für eine bestimmte Reihe von Aufgaben verantwortlich sind. Jeder Workstream ist unabhängig, unterstützt aber die anderen Workstreams

im Projekt. Der Portfolio-Workstream ist beispielsweise für die Priorisierung von Anwendungen, die Wellenplanung und die Erfassung von Migrationsmetadaten verantwortlich. Der Portfolio-Workstream liefert diese Komponenten an den Migrations-Workstream, der dann die Server und Anwendungen migriert.

WURM

Sehen Sie einmal schreiben, viele lesen.

WQF

Siehe AWS Workload-Qualifizierungsrahmen.

einmal schreiben, viele lesen (WORM)

Ein Speichermodell, das Daten ein einziges Mal schreibt und verhindert, dass die Daten gelöscht oder geändert werden. Autorisierte Benutzer können die Daten so oft wie nötig lesen, aber sie können sie nicht ändern. Diese Datenspeicherinfrastruktur gilt als unveränderlich.

Z

Zero-Day-Exploit

Ein Angriff, in der Regel Malware, der eine Zero-Day-Sicherheitslücke ausnutzt.

Zero-Day-Sicherheitslücke

Ein unfehlbarer Fehler oder eine Sicherheitslücke in einem Produktionssystem. Bedrohungsakteure können diese Art von Sicherheitslücke nutzen, um das System anzugreifen. Entwickler werden aufgrund des Angriffs häufig auf die Sicherheitsanfälligkeit aufmerksam.

Eingabeaufforderung ohne Zwischenfälle

Bereitstellung von Anweisungen für die Ausführung einer Aufgabe an einen LLM, jedoch ohne Beispiele (Schnappschüsse), die ihm als Orientierungshilfe dienen könnten. Der LLM muss sein vortrainiertes Wissen einsetzen, um die Aufgabe zu bewältigen. Die Effektivität von Zero-Shot Prompting hängt von der Komplexität der Aufgabe und der Qualität der Aufforderung ab. Siehe auch Few-Shot-Prompting.

Zombie-Anwendung

Eine Anwendung, deren durchschnittliche CPU- und Arbeitsspeichernutzung unter 5 Prozent liegt. In einem Migrationsprojekt ist es üblich, diese Anwendungen außer Betrieb zu nehmen.

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.