Das Tool CI/CD litmus test: Is your pipeline fully CI/CD?

AWS Präskriptive Leitlinien

AWS Präskriptive Leitlinien: Das Tool CI/CD litmus test: Is your pipeline fully CI/CD?

Table of Contents

Das Tool CI/CD litmus test: Is your pipeline fully CI/CD?

Steven Guggenheimer und Ananya Koduri, Amazon Web Services ()AWS

August 2023 (Dokumentverlauf)

Ist Ihre Pipeline automatisiert? Es ist eine einfache Frage, aber viele Unternehmen gehen die Antwort zu einfach an. Die Antwort ist viel komplizierter als ein Ja oder Nein.

Technologische Innovationen finden ständig statt, und manchmal kann es für Unternehmen schwierig sein, Schritt zu halten. Ist dieses neue Ding eine Modeerscheinung oder ist es das nächste große Ding? Sollte ich meine derzeitigen Praktiken überarbeiten oder sollte ich warten? Wenn klar wird, dass etwas tatsächlich das nächste große Ding ist, kann es oft passieren, dass Sie aufholen. Continuous Integration and Continuous Delivery (CI/CD) ist gekommen, um zu bleiben, aber das war nicht immer so. Viele Menschen haben lange gebraucht, um überzeugt zu werden, und manche Menschen brauchen noch mehr Überzeugungsarbeit.

CI/CD is the process of automating the source, build, test, staging, and production stages of the software release process, and it is commonly described as a pipeline. Today, the cost savings and speed of CI/CDAutomatisierungen haben die meisten Organisationen von ihrem Wert überzeugt. Die Umstellung auf diesen neuen Ansatz ist jedoch keine leichte Aufgabe. Sie müssen sicherstellen, dass Ihre Mitarbeiter über die richtige Ausbildung verfügen, Sie müssen einige Ressourcen aufrüsten, und dann müssen Sie testen, testen, testen. Es gibt viel zu tun. In den meisten Fällen sollten Sie diese Änderungen schrittweise vornehmen, um Ihr Unternehmen bei der Anpassung zu unterstützen.

Der Zweck dieses Dokuments besteht darin, zu definieren, was es bedeutet, einen vollständigen CI/CD-Prozess zu haben. Es bietet ein Tool zur Bewertung Ihrer eigenen Prozesse und zeigt einen Weg in die Zukunft für Prozesse auf, die es noch nicht gibt. Dieser Weg in die Zukunft ist selten eine Umstellung über Nacht. Diese Prozesse sind komplex und hängen von vielen Faktoren ab, darunter den aktuellen Fähigkeiten der Mitarbeiter und den aktuellen Infrastrukturanforderungen. Wir empfehlen Ihnen, Prioritäten zu setzen und kleine, schrittweise Änderungen vorzunehmen.

Ziele

Im Folgenden sind die potenziellen Vorteile der Umsetzung der Empfehlungen in diesem Leitfaden aufgeführt:

- Effizienz — Ein vollständiger CI/CD-Bereitstellungsprozess kann die Komplexität, die Arbeitslast und die unzähligen Stunden reduzieren, die für das Debuggen, die Durchführung manueller

Prozesse und die Wartung aufgewendet werden müssen. Weitere Informationen finden Sie unter Vorteile der kontinuierlichen Auslieferung. Einem TechAhead Blogbeitrag zufolge kann die Implementierung des CI/CD-Prozesses zu geschätzten Einsparungen von 20% an Zeit, Aufwand und Ressourcen führen.

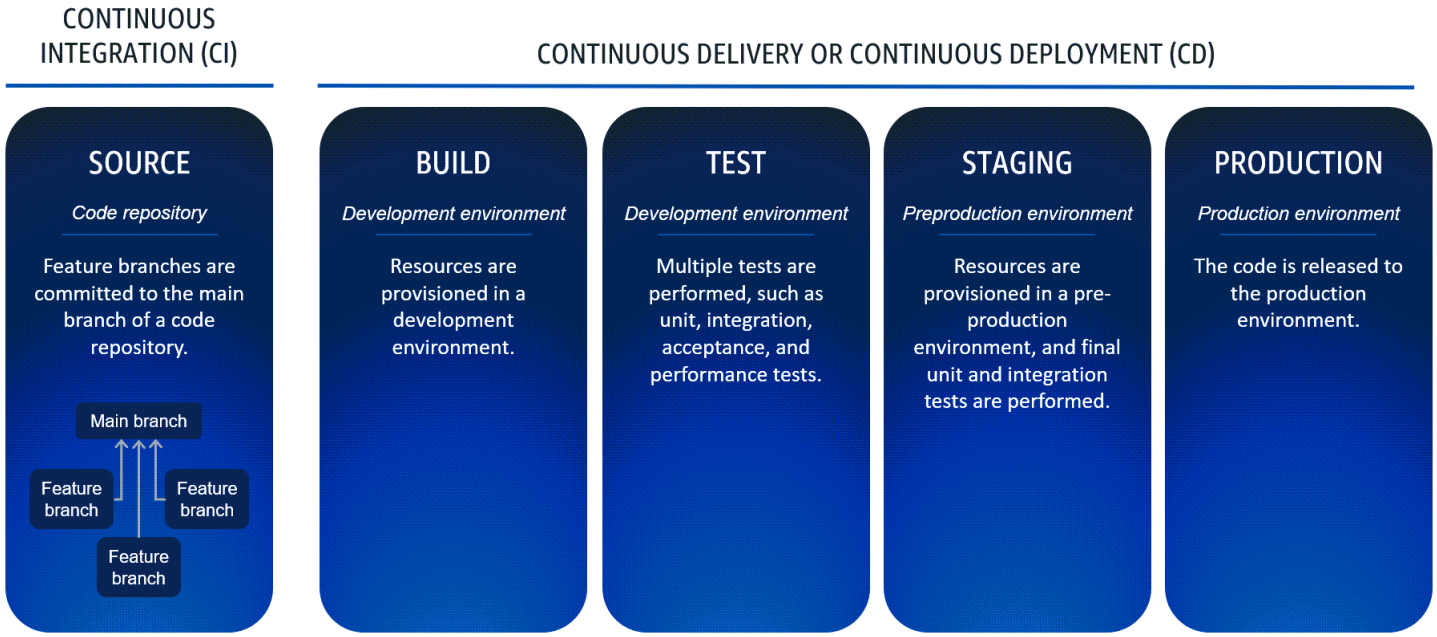
- Kostensenkung — Laut einem Bericht von Forbes Insight „sind sich drei von vier Führungskräften einig, dass der Zeit-, Geld- und Ressourcenaufwand für die laufende Wartung und Verwaltung — im Vergleich zu neuen Projektentwicklungen oder neuen Initiativen — die allgemeine Wettbewerbsfähigkeit ihres Unternehmens beeinträchtigt.“ Je kürzer der Entwicklungszyklus, desto höher sind die Chancen, dass Ihr Unternehmen ehrgeizige time-to-market Ziele erreichen und die richtigen Chancen zur richtigen Zeit nutzen kann.
- Geschwindigkeit — In der Regel trägt eine vollständige CI/CD pipeline is able to release software changes to customers within a few hours. Especially in cases with quick fault isolations and small patch pushes, the CI/CD Pipeline dazu bei, die mittlere Wiederherstellungszeit (MTTR) zu verkürzen. Weitere Informationen finden Sie unter Reduzierung der MTTR.
- Sicherheit — Vollständige CI/CD pipelines also secure the release process by reducing the possible entry points for attacks and reducing the risk of human error. The security gains that come with fully automated CI/CD Pipelines tragen dazu bei, die kostspieligen Folgen von Datenschutzverletzungen, Serviceausfällen und mehr zu vermeiden.
- Geringere Fluktuation — Entwickler sind zufriedener, wenn sie mehr Zeit mit der Entwicklung großartiger Funktionen verbringen können und weniger Zeit in einem endlosen Wartungs- und Debugging-Zyklus verbringen müssen. Für Unternehmen bedeutet dies, Top-Talente zu gewinnen und über einen längeren Zeitraum zu binden.
- Erstklassiger Code — Entwickler veröffentlichen Code in kleinen Batches in einem gemeinsamen Repository, sodass sie parallel Tests durchführen können (BrowserStack Blogbeitrag). Anstatt isoliert zu arbeiten, teilen sie ihre Builds häufig mit dem Team und arbeiten zusammen, um kritische Fehler zu identifizieren. Dies bietet Unterstützung für Entwickler, wodurch verhindert wird, dass schlechter Code in die Produktion gelangt. Die Support durch Kollegen aus den Entwicklern trägt zu qualitativ hochwertigen Releases bei und fördert das Unternehmenswachstum.
- Wartung — Wartung und Updates sind ein entscheidender Bestandteil der Entwicklung eines großartigen Produkts. Schalten Sie das System jedoch nicht zu Spitzenzeiten aus. Sie können eine CI/CD-Pipeline verwenden, um Wartungsarbeiten zu Zeiten mit geringer Auslastung durchzuführen, um Ausfallzeiten und Leistungseinbußen zu minimieren.

CI/CD verstehen

Continuous Integration and Continuous Delivery (CI/CD) ist der Prozess der Automatisierung des Lebenszyklus von Softwareversionen. In einigen Fällen kann das D in CI/CD auch Bereitstellung bedeuten. Der Unterschied zwischen Continuous Delivery und Continuous Deployment tritt auf, wenn Sie eine Änderung an der Produktionsumgebung veröffentlichen. Bei Continuous Delivery ist eine manuelle Genehmigung erforderlich, bevor Änderungen an der Produktion vorgenommen werden. Die kontinuierliche Bereitstellung ermöglicht einen unterbrechungsfreien Ablauf der gesamten Pipeline, und es sind keine ausdrücklichen Genehmigungen erforderlich. Da in dieser Strategie allgemeine CI/CD-Konzepte behandelt werden, gelten die bereitgestellten Empfehlungen und Informationen sowohl für die kontinuierliche Bereitstellung als auch für die kontinuierliche Bereitstellung.

CI/CD automates much or all of the manual processes traditionally required to get new code from a commit into production. A CI/CD pipeline encompasses the source, build, test, staging, and production stages. In each stage, the CI/CD pipelines provisions any infrastructure that is needed to deploy or test the code. By using a CI/CDIn der Pipeline können Entwicklungsteams Änderungen am Code vornehmen, die dann automatisch getestet und zur Bereitstellung weitergeleitet werden.

Sehen wir uns die grundlegenden CI/CD process before discussing some of the ways that you can, knowingly or unknowingly, deviate from being fully CI/CD. The following diagram shows the CI/CD Phasen und Aktivitäten in jeder Phase an.



Über kontinuierliche Integration

Die kontinuierliche Integration erfolgt in einem Code-Repository, z. B. einem Git-Repository in GitHub. Sie behandeln einen einzelnen Hauptzweig als Informationsquelle für die Codebasis und erstellen kurzlebige Zweige für die Feature-Entwicklung. Sie integrieren einen Feature-Branch in den Hauptzweig, wenn Sie bereit sind, das Feature in höheren Umgebungen bereitzustellen. Feature-Branches werden niemals direkt in höheren Umgebungen bereitgestellt. Weitere Informationen finden Sie unter Trunk-basierter Ansatz in diesem Handbuch.

Kontinuierlicher Integrationsprozess

1. Der Entwickler erstellt aus dem Hauptzweig einen neuen Zweig.
2. Der Entwickler nimmt Änderungen vor und erstellt und testet lokal.
3. Wenn die Änderungen fertig sind, erstellt der Entwickler eine Pull-Anfrage (GitHub Dokumentation) mit dem Hauptzweig als Ziel.
4. Der Code wird überprüft.
5. Wenn der Code genehmigt ist, wird er mit dem Hauptzweig zusammengeführt.

Über kontinuierliche Lieferung

Continuous Delivery findet in isolierten Umgebungen statt, z. B. in Entwicklungs- und Produktionsumgebungen. Die Aktionen, die in den einzelnen Umgebungen ausgeführt werden, können variieren. Oft wird eine der ersten Phasen verwendet, um Aktualisierungen an der Pipeline selbst vorzunehmen, bevor der Vorgang fortgesetzt wird. Das Endergebnis der Bereitstellung ist, dass jede Umgebung mit den neuesten Änderungen aktualisiert wird. Die Anzahl der Entwicklungsumgebungen zum Erstellen und Testen variiert ebenfalls, wir empfehlen jedoch, mindestens zwei zu verwenden. In der Pipeline wird jede Umgebung in der Reihenfolge ihrer Bedeutung aktualisiert, bis die wichtigste Umgebung, die Produktionsumgebung, abgeschlossen wird.

Kontinuierlicher Lieferprozess

Der Continuous-Delivery-Teil der Pipeline wird initiiert, indem der Code aus dem Hauptzweig des Quell-Repositorys abgerufen und an die Build-Phase übergeben wird. Das Dokument Infrastructure as Code (IaC) für das Repository beschreibt die Aufgaben, die in den einzelnen Phasen ausgeführt werden. Die Verwendung eines IaC-Dokuments ist zwar nicht verpflichtend, es wird jedoch dringend empfohlen, einen IaC-Dienst oder ein IaC-Tool wie AWS CloudFormationoder AWS Cloud Development Kit (AWS CDK)zu verwenden. Zu den gängigsten Schritten gehören:

1. Komponententests
2. Code erstellen
3. Bereitstellung von Ressourcen
4. Integrationstests

Wenn in einer Phase der Pipeline Fehler auftreten oder Tests fehlschlagen, wird die aktuelle Phase auf ihren vorherigen Status zurückgesetzt und die Pipeline wird beendet. Nachfolgende Änderungen müssen im Code-Repository beginnen und den vollständigen CI/CD-Prozess durchlaufen.

Tests für CI/CD-Pipelines

Die beiden Arten automatisierter Tests, auf die in Bereitstellungspipelines häufig Bezug genommen wird, sind Komponententests und Integrationstests. Es gibt jedoch viele Arten von Tests, die Sie auf einer Codebasis und in der Entwicklungsumgebung ausführen können. Die AWS Deployment Pipeline Reference Architecture definiert die folgenden Testtypen:

- Komponententest — Diese Tests erstellen Anwendungscode und führen ihn aus, um zu überprüfen, ob er erwartungsgemäß funktioniert. Sie simulieren alle externen Abhängigkeiten, die in der Codebasis verwendet werden. Beispiele für Unit-Test-Tools sind JUnitJest und Pytest.
- Integrationstest — Diese Tests verifizieren, dass die Anwendung die technischen Anforderungen erfüllt, indem sie anhand einer bereitgestellten Testumgebung getestet werden. Beispiele für Integrationstesttools sind Cucumber, vRest NG und Integ-Tests (for). AWS CDK
- Akzeptanztest — Bei diesen Tests wird anhand einer bereitgestellten Testumgebung überprüft, ob die Anwendung die Benutzeranforderungen erfüllt. Zu den Tools für Akzeptanztests gehören beispielsweise Cypress und Selenium.
- Synthetischer Test — Diese Tests werden kontinuierlich im Hintergrund ausgeführt, um Traffic zu generieren und zu überprüfen, ob das System fehlerfrei ist. Beispiele für synthetische Testtools sind Amazon CloudWatch Synthetics und Dynatrace Synthetic Monitoring.
- Leistungstest — Diese Tests simulieren die Produktionskapazität. Sie ermitteln, ob die Anwendung die Leistungsanforderungen erfüllt, und vergleichen die Kennzahlen mit der Leistung in der Vergangenheit. Zu den Tools für Leistungstests gehören beispielsweise Apache JMeter, Locust und Gatling.
- Resilienztest — Diese Tests, auch Chaostests genannt, führen zu Fehlern in Umgebungen, um Risikobereiche zu identifizieren. Perioden, in denen die Fehler auftreten, werden dann mit Perioden

ohne Fehler verglichen. Zu den Tools für Resilienz-Tests gehören beispielsweise AWS Fault Injection ServiceGremlin.

- Statischer Anwendungssicherheitstest (SAST) — Diese Tests analysieren Code auf Sicherheitsverletzungen wie SQL-Injection oder Cross-Site Scripting (XSS). Beispiele für SAST-Tools sind Amazon CodeGuru und SonarQubeCheckmarx.
- Dynamischer Anwendungssicherheitstest (DAST) — Diese Tests werden auch als Penetrationstests oder Penetrationstests bezeichnet. Sie identifizieren Sicherheitslücken wie SQL-Injection oder XSS in einer bereitgestellten Testumgebung. Beispiele für DAST-Tools sind Zed Attack Proxy (ZAP) und HCL. AppScan Weitere Informationen finden Sie unter Penetrationstests.

Nicht alle CI/CD-Pipelines führen alle diese Tests durch. Eine Pipeline sollte jedoch mindestens Unit-Tests und SAST-Tests auf der Codebasis sowie Integrations- und Akzeptanztests in einer Testumgebung ausführen.

Metriken für CI/CD-Pipelines

Gemäß der AWS Deployment Pipeline Reference Architecture sollten Sie mindestens die folgenden vier Metriken für CI/CD-Pipelines verfolgen:

- Vorlaufzeit — Die durchschnittliche Zeit, die ein einzelnes Commit benötigt, um in die Produktion überzugehen. Wir empfehlen, je nach Anwendungsfall eine Vorlaufzeit zwischen 1 Stunde und 1 Tag anzustreben.
- Bereitstellungshäufigkeit — Die Anzahl der Produktionsbereitstellungen innerhalb eines bestimmten Zeitraums. Wir empfehlen, die Bereitstellungshäufigkeit je nach Anwendungsfall zwischen mehrmals täglich und zweimal pro Woche festzulegen.
- Mittlere Zeit zwischen Ausfällen (MTBF) — Die durchschnittliche Zeit zwischen dem Start einer erfolgreichen Pipeline und dem Start einer ausgefallenen Pipeline. Wir empfehlen, eine möglichst hohe MTBF anzustreben. Weitere Informationen finden Sie unter Erhöhung der MTBF.
- Mittlere Zeit bis zur Wiederherstellung (MTTR) — Die durchschnittliche Zeit zwischen dem Start einer ausgefallenen Pipeline und dem Start der nächsten erfolgreichen Pipeline. Wir empfehlen, eine MTTR anzustreben, die so niedrig wie möglich ist. Weitere Informationen finden Sie unter MTTR reduzieren.

Diese Kennzahlen helfen Teams dabei, ihre Fortschritte auf dem Weg zu einer vollständigen CI/CD-Implementierung nachzuverfolgen. Die Teams sollten offene Diskussionen mit den Stakeholdern der

Organisation darüber führen, was die optimalen Ziele sein sollten. Situationen und Bedürfnisse sind von Organisation zu Organisation und sogar von Team zu Team sehr unterschiedlich.

Es ist sehr wichtig, sich daran zu erinnern, dass schnelle, drastische Veränderungen in der Regel das Risiko erhöhen, dass Probleme auftreten. Setzen Sie sich Ziele, um kleine, schrittweise Verbesserungen anzustreben. Eine übliche optimale Vorlaufzeit für vollständige CI/CD-Pipelines beträgt weniger als 3 Stunden. Ein Team, das mit einer Vorlaufzeit von 5,2 Tagen beginnt, sollte eine Reduzierung um einen Tag alle paar Wochen anstreben. Sobald dieses Team eine Vorlaufzeit von einem Tag oder weniger erreicht hat, kann es dort mehrere Monate bleiben und nur dann zu einer aggressiveren Vorlaufzeit übergehen, wenn das Team und die Interessengruppen der Organisation dies für notwendig halten.

Inwieweit unterscheiden sich CI/CD-Prozesse

CI/CD-Pipelines verwenden einen modernen, auf Trunk basierenden Workflow, bei dem Entwickler kleine, häufige Updates zu einem Hauptzweig (oder Trunk) zusammenführen, der über den CD-Teil der CI/CD-Pipeline erstellt und getestet wird. Dieser Workflow hat den Gitflow-Workflow ersetzt, bei dem Entwicklungs- und Release-Zweige durch einen Release-Zeitplan getrennt sind. In vielen Organisationen ist Gitflow immer noch eine beliebte Methode zur Versionskontrolle und -bereitstellung. Inzwischen gilt es jedoch als veraltet, und es kann schwierig sein, es in eine CI/CD-Pipeline zu integrieren.

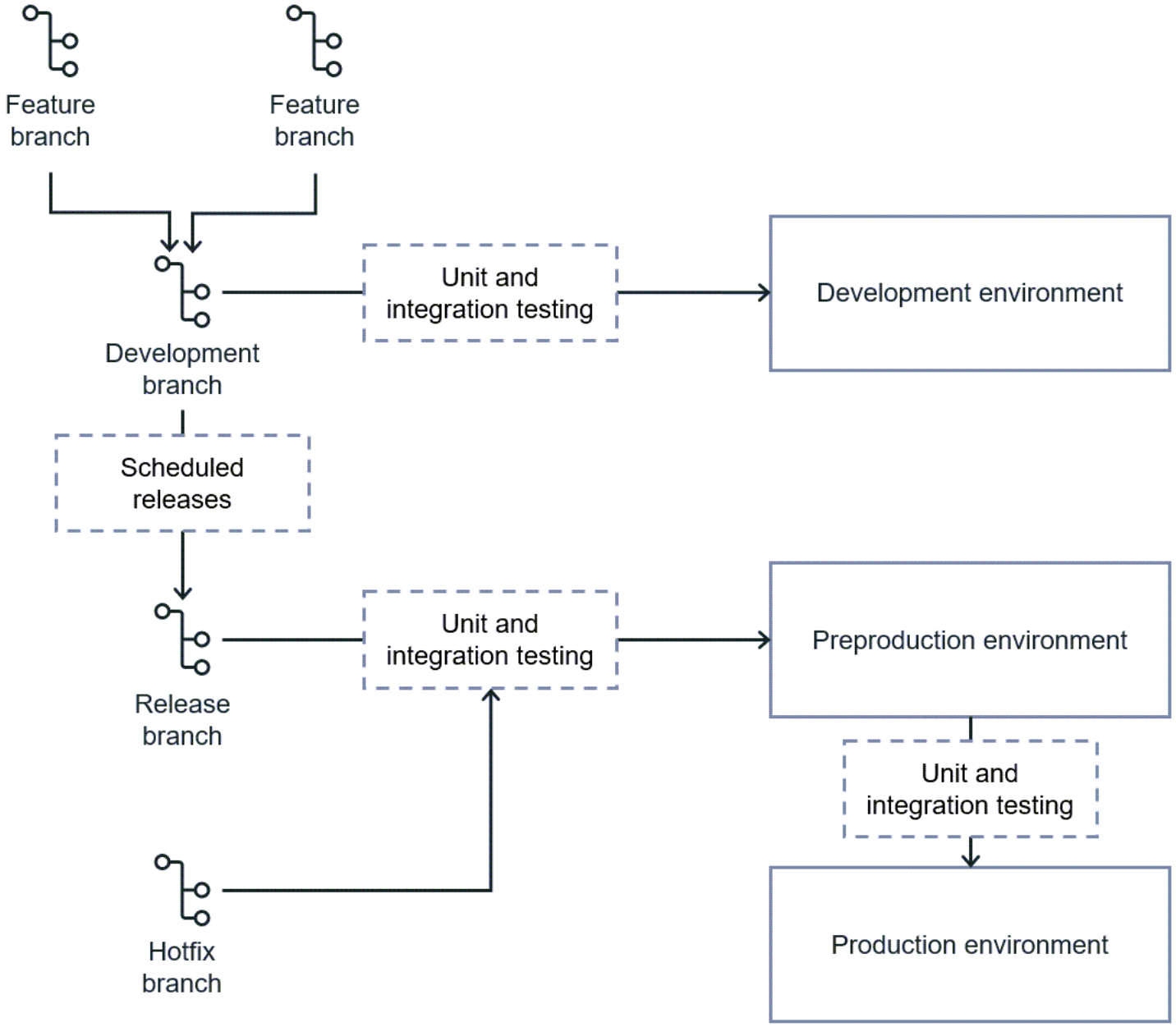
Für viele Unternehmen war der Übergang von einem Gitflow-Workflow zu einem Trunk-basierten Workflow unvollständig. Das Ergebnis ist, dass sie irgendwo auf dem Weg stecken bleiben und nie vollständig auf CI/CD migrieren. Irgendwie klammern sich ihre Pipelines am Ende an bestimmte Überreste des veralteten Workflows, die sich in einem Übergangszustand zwischen Vergangenheit und Gegenwart befinden. Sehen Sie sich die Unterschiede in den Git-Workflows an und erfahren Sie dann, wie sich die Verwendung eines Legacy-Workflows auf Folgendes auswirken kann:

- Integrität der Umwelt
- Versionen
- Sicherheit

Um es einfacher zu machen, die Überreste eines alten Git-Workflows in einer modernen Konfiguration zu identifizieren, vergleichen wir Gitflow mit dem modernen, Trunk-basierten Ansatz.

Gitflow-Ansatz

Das folgende Bild zeigt einen Gitflow-Workflow. Der Gitflow-Ansatz verwendet mehrere Branches, um mehrere verschiedene Versionen des Codes gleichzeitig zu verfolgen. Sie planen die Veröffentlichung von Updates für eine Anwendung für einen bestimmten Zeitpunkt in der future, während die Entwickler noch an der aktuellen Version des Codes arbeiten. Trunk-basierte Repositorys können Feature-Flags verwenden, um dies zu erreichen, aber sie sind standardmäßig in Gitflow integriert.



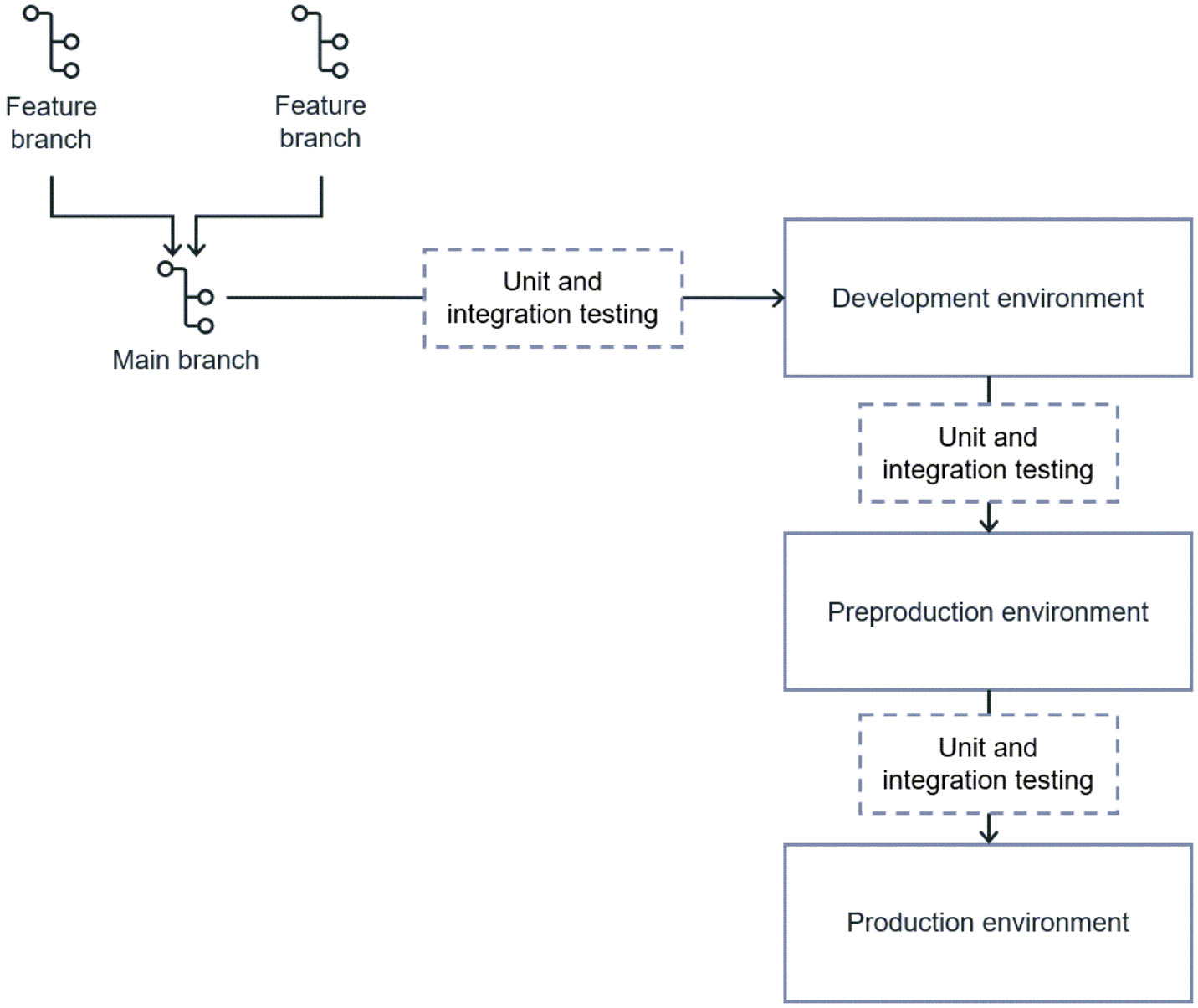
Ein Ergebnis des Gitflow-Ansatzes ist, dass die Anwendungsumgebungen normalerweise nicht synchron sind. In einer Gitflow-Standardimplementierung spiegeln die Entwicklungsumgebungen den aktuellen Status des Codes wider, während die Vorproduktions- und Produktionsumgebungen auf dem Stand der Codebasis aus der neuesten Version stehen.

Dies verkompliziert die Situation, wenn ein Fehler in der Produktionsumgebung auftritt, da die Codebasis, in der die Entwickler arbeiten, nicht mit der Produktion zusammengeführt werden kann, ohne unveröffentlichte Funktionen offenzulegen. Gitflow geht mit dieser Situation um, indem es einen Hotfix verwendet. Aus dem Release-Zweig wird ein Hotfix-Branch erstellt und dann direkt in

den oberen Umgebungen bereitgestellt. Der Hotfix-Zweig wird dann mit dem Entwicklungszweig zusammengeführt, um den Code auf dem neuesten Stand zu halten.

Trunk-basierter Ansatz

Die folgende Abbildung zeigt einen Trunk-basierten Workflow. In einem Trunk-basierten Workflow erstellen und testen Entwickler Features lokal in einem Feature-Branch und führen diese Änderungen dann im Hauptzweig zusammen. Der Hauptzweig wird dann sequentiell für die Entwicklungs-, Vorproduktions- und Produktionsumgebungen erstellt. Einheiten- und Integrationstests finden zwischen den einzelnen Umgebungen statt.



Bei Verwendung dieses Workflows verwenden alle Umgebungen dieselbe Codebasis. Für die oberen Umgebungen ist kein Hotfix-Zweig erforderlich, da Sie Änderungen im Hauptzweig implementieren können, ohne unveröffentlichte Funktionen verfügbar zu machen. Es wird immer davon ausgegangen, dass der Hauptzweig stabil, fehlerfrei und bereit zur Veröffentlichung ist. Auf diese Weise können Sie ihn als Quelle für eine CI/CD-Pipeline integrieren, mit der Ihre Codebasis automatisch getestet und in allen Umgebungen in Ihrer Pipeline bereitgestellt werden kann.

Vorteile eines Trunk-basierten Ansatzes für die Integrität der Umgebung

Wie viele Entwickler wissen, kann eine Änderung im Code manchmal einen Schmetterlingseffekt erzeugen (Artikel von American Scientist), bei dem eine kleine Abweichung, die scheinbar nichts miteinander zu tun hat, eine Kettenreaktion auslöst, die zu unerwarteten Ergebnissen führt. Entwickler müssen dann alles untersuchen, um die Ursache zu finden.

Wenn Wissenschaftler ein Experiment durchführen, teilen sie die Testpersonen in zwei Gruppen ein: die Versuchsgruppe und die Kontrollgruppe. Die Absicht ist, die Versuchsgruppe und die Kontrollgruppe bis auf das, was im Experiment getestet wird, völlig identisch zu machen. Wenn in der Versuchsgruppe etwas passiert, was in der Kontrollgruppe nicht passiert, kann die einzige Ursache das getestete Ding sein.

Stellen Sie sich die Änderungen in einer Bereitstellung als Versuchsgruppe und jede Umgebung als separate Kontrollgruppen vor. Die Ergebnisse von Tests in einer niedrigeren Umgebung sind nur dann zuverlässig, wenn die Kontrollen dieselben sind wie in einer oberen Umgebung. Je mehr die Umgebungen abweichen, desto größer ist die Wahrscheinlichkeit, dass Fehler in den oberen Umgebungen entdeckt werden. Mit anderen Worten, wenn die Codeänderungen in der Produktion fehlschlagen, wäre es uns viel lieber, wenn sie zuerst in der Betaversion fehlschlagen, damit sie nie in Produktion gehen. Aus diesem Grund sollten alle Anstrengungen unternommen werden, um jede Umgebung, von der niedrigsten Testumgebung bis hin zur Produktion selbst, synchron zu halten. Dies wird als Umgebungsintegrität bezeichnet.

Das Ziel eines vollständigen CI/CD-Prozesses besteht darin, Probleme so früh wie möglich zu erkennen. Durch die Wahrung der Umgebungsintegrität mithilfe eines Trunk-basierten Ansatzes können Hotfixes praktisch überflüssig werden. In einem Trunk-basierten Workflow kommt es selten vor, dass ein Problem zuerst in der Produktionsumgebung auftritt.

Bei einem Gitflow-Ansatz wird ein Hotfix, nachdem er direkt in höheren Umgebungen bereitgestellt wurde, dem Entwicklungszweig hinzugefügt. Dadurch bleibt das Update für future Versionen

erhalten. Der Hotfix wurde jedoch direkt auf der Grundlage des aktuellen Status der Anwendung entwickelt und getestet. Selbst wenn der Hotfix in der Produktionsumgebung einwandfrei funktioniert, besteht die Möglichkeit, dass Probleme auftreten, wenn er mit den neueren Funktionen in der Entwicklungsabteilung interagiert. Da die Bereitstellung eines Hotfixes für einen Hotfix normalerweise nicht wünschenswert ist, bedeutet dies, dass Entwickler zusätzliche Zeit damit verbringen, den Hotfix in die Entwicklungsumgebung nachzurüsten. In vielen Fällen kann dies zu erheblichen technischen Schulden führen und die Gesamtstabilität der Entwicklungsumgebung beeinträchtigen.

Wenn in einer Umgebung ein Fehler auftritt, werden alle Änderungen rückgängig gemacht, sodass die Umgebung in ihren vorherigen Zustand zurückversetzt wird. Bei jeder Änderung an einer Codebasis sollte die Pipeline von der allerersten Phase an erneut gestartet werden. Wenn in der Produktionsumgebung ein Problem auftritt, sollte das Update auch die gesamte Pipeline durchlaufen. Die zusätzliche Zeit, die benötigt wird, um die niedrigeren Umgebungen zu durcharbeiten, ist im Vergleich zu den Problemen, die mit diesem Ansatz vermieden werden, in der Regel vernachlässigbar. Da der gesamte Zweck der unteren Umgebungen darin besteht, Fehler zu catch bevor sie in die Produktion gelangen, ist die Umgehung dieser Umgebungen durch einen Gitflow-Ansatz ein ineffizientes und unnötiges Risiko.

Nutzen Sie die Vorteile eines Trunk-basierten Ansatzes

Ein Hotfix ist häufig erforderlich, weil in einem älteren Workflow der Status der Anwendung, an der die Entwickler arbeiten, möglicherweise mehrere unveröffentlichte Funktionen enthält, die noch nicht in der Produktion verfügbar sind. Die Produktionsumgebung und die Entwicklungsumgebung werden erst synchronisiert, wenn eine geplante Veröffentlichung veröffentlicht wird, und dann beginnen sie sofort wieder voneinander abzuweichen, bis die nächste geplante Version erscheint.

Die Möglichkeit, geplante Releases innerhalb eines vollständigen CI/CD process. You can delay the release of code to production by using feature flags. However, a fully CI/CD Prozesses zu haben, ermöglicht mehr Flexibilität, da geplante Releases überflüssig werden. Schließlich ist kontinuierlich ein Schlüsselwort in CI/CD, und das deutet darauf hin, dass Änderungen veröffentlicht werden, sobald sie fertig sind. Vermeiden Sie die Beibehaltung einer separaten Release-Umgebung, die fast immer nicht mit den Testumgebungen auf niedrigerem Niveau synchron ist.

Wenn eine Pipeline nicht vollständig CI/CD ist, tritt die Divergenz zwischen den oberen und unteren Umgebungen normalerweise auf Zweigstellenebene auf. Entwickler arbeiten in einem Entwicklungszweig und verwalten einen separaten Release-Zweig, der nur aktualisiert wird, wenn es Zeit für eine geplante Veröffentlichung ist. Da der Release-Zweig und der Entwicklungszweig voneinander abweichen, können weitere Komplikationen auftreten.

Wenn Entwickler in der Entwicklungsabteilung arbeiten und sich an einen Anwendungsstatus gewöhnen, der dem in der Produktion weit voraus ist, müssen sie sich nicht nur an den Produktionsstatus anpassen, sondern müssen sich auch jedes Mal, wenn dort ein Problem auftritt, wieder an den Produktionsstatus anpassen. Der Stand der Entwicklungsabteilung könnte der Produktion um viele Funktionen voraus sein. Wenn Entwickler täglich in dieser Branche arbeiten, ist es schwierig, sich daran zu erinnern, was für die Produktion freigegeben ist und was nicht. Dies erhöht das Risiko, dass neue Fehler eingeführt werden, während andere Fehler behoben werden. Dieses Ergebnis ist ein scheinbar endloser Zyklus von Korrekturen, die die Zeitpläne verlängern und Feature-Releases um Wochen, Monate oder sogar Jahre verzögern.

Sicherheitsvorteile eines Trunk-basierten Ansatzes

Ein vollständiger CI/CD-Prozess bietet einen vollautomatischen Ansatz für die Bereitstellung, der sich auf eine zentrale Informationsquelle stützt. Die Pipeline hat einen einzigen Zugangspunkt. Softwareupdates werden zu Beginn in die Pipeline aufgenommen und unverändert von einer Umgebung zur nächsten weitergeleitet. Wenn in einer Phase der Pipeline ein Problem entdeckt wird, müssen die Codeänderungen, mit denen das Problem behoben wird, denselben Prozess durchlaufen und mit der ersten Phase beginnen. Durch die Reduzierung der Eintrittspunkte in einer Pipeline werden auch die Möglichkeiten reduziert, wie Sicherheitslücken in die Pipeline eingebracht werden können.

Da der Eintrittspunkt zudem so weit wie möglich von der Produktionsumgebung entfernt ist, wird die Wahrscheinlichkeit, dass Sicherheitslücken in die Produktionsumgebung gelangen, drastisch reduziert. Wenn Sie einen manuellen Genehmigungsprozess in einer vollständigen CI/CD-Pipeline implementieren, können Sie immer noch entscheiden, ob Änderungen in die nächste Umgebung übertragen werden oder nicht. Der Entscheidungsträger ist nicht unbedingt dieselbe Person, die Änderungen vornimmt. Dadurch werden die Verantwortlichkeiten des Implementierers von Codeänderungen und des Genehmigers dieser Änderungen getrennt. Dadurch wird es auch für weniger technisch versierte Unternehmensleiter praktikabler, die Rolle des Genehmigers wahrzunehmen.

Und schließlich hilft Ihnen der zentrale Zugangspunkt dabei, den Schreibzugriff auf die Benutzeroberflächenkonsole (UI) der Produktionsumgebung auf einige oder gar keine Benutzer zu beschränken. Indem Sie die Anzahl der Benutzer reduzieren, die manuelle Änderungen an der Konsole vornehmen können, verringern Sie das Risiko von Sicherheitsereignissen. Die Möglichkeit, die Konsole in der Produktionsumgebung manuell zu verwalten, ist bei älteren Workflows weitaus wichtiger als bei einem automatisierten CI/CD-Ansatz. Diese manuellen Änderungen sind schwieriger

nachzuverfolgen, zu überprüfen und zu testen. Sie werden normalerweise durchgeführt, um Zeit zu sparen, aber auf lange Sicht führen sie zu einer erheblichen technischen Belastung des Projekts.

Sicherheitsprobleme auf Konsolen werden nicht unbedingt von böswilligen Akteuren verursacht. Viele der Probleme, die in der Konsole auftreten, sind unbeabsichtigt. Versehentliche Sicherheitslücken sind sehr häufig und haben zur Verbreitung des Zero-Trust-Sicherheitsmodells geführt. Dieses Modell geht teilweise davon aus, dass Sicherheitsunfälle weniger wahrscheinlich sind, wenn selbst interne Mitarbeiter so wenig Zugriff wie möglich haben, was auch als Least-Privilege-Berechtigungen bezeichnet wird. Wenn die Integrität der Produktionsumgebung gewahrt wird, indem alle Prozesse auf eine automatisierte Pipeline beschränkt werden, wird das Risiko von Sicherheitsproblemen im Zusammenhang mit der Konsole praktisch ausgeschlossen.

Lackmustest für CI/CD-Pipelines

In der Chemie ist Lackmuspapier ein dünner Papierstreifen, der mit einem speziellen roten oder blauen Farbstoff behandelt wurde, der zur Bestimmung des Säuregehalts einer Substanz verwendet wird. Eine Säure färbt blaues Lackmuspapier rot, eine Base färbt rotes Lackmuspapier blau, und neutrale Substanzen beeinflussen die Farbe des Papiers überhaupt nicht.

Lackmuspapier bestimmt den Säuregehalt, indem es den pH-Wert einer Substanz misst. Wenn ein pH-Wert höher als 8 ist, ist er sauer; liegt er unter 5, ist er basisch; und liegt er zwischen 5 und 8, ist er neutral. In ähnlicher Weise hilft Ihnen der CI/CD-Lackmustest dabei, den CI/CD-Wert Ihrer Pipeline zu messen.

Um zu testen, ob Ihre Pipeline vollständig CI/CD ist

1. Beginnen Sie mit einer Punktzahl von 0.
2. Beantworten Sie jede der folgenden Fragen und addieren Sie 1 zu Ihrer Punktzahl für jedes Mal, wenn Sie mit Ja antworten:

    - Haben unsere Repositorys jeweils genau einen Hauptzweig, der für die Bereitstellung in Umgebungen verwendet wird?
    - Übergeben wir Code häufig an den Hauptzweig und vermeiden so lange laufende Feature-Branches?
    - Hat unsere Pipeline einen einzigen Einstiegspunkt? Mit anderen Worten, ruft unsere Pipeline den Code genau einmal aus jedem Repository ab?
    - Haben wir mehr als eine Bereitstellungsumgebung?
    - Wenn die Pipeline nicht läuft, sind unsere oberen und unteren Umgebungen im Allgemeinen synchron?
    - Führen wir vor der Bereitstellung Tests mit Code durch?
    - Führen wir Tests in einer Umgebung durch, bevor wir zur nächsten Umgebung wechseln?
    - Führt unsere Pipeline ein vollständiges Rollback durch und wird sie nach einem Ausfall beendet?
    - Wird unsere Pipeline bei der Wiederherstellung nach einem Ausfall vom ersten Schritt an neu gestartet?
    - Gehen wir bei der Behebung von Fehlern in der Produktion genauso vor wie bei der Freigabe von Funktionen für die Produktion?

- Verwenden wir irgendeine Form von Infrastructure-as-Code-Vorlagen (IaC), um Code bereitzustellen?

3. Beantworten Sie jede der folgenden Fragen und addieren Sie für jedes Mal, wenn Sie mit Nein antworten, 1 zu Ihrer Punktzahl:

   - Implementieren wir jemals direkt in einer Bereitstellungsumgebung von anderen Zweigstellen als der Hauptzweige aus?
   - Implementieren wir jemals direkt von einer Filiale aus in eine übergeordnete Umgebung oder eine Produktionsumgebung?
   - Finden wir häufig Fehler in höheren Umgebungen, die in niedrigeren Umgebungen nicht vorhanden waren?
   - Umgehen wir während einer Bereitstellung jemals niedrigere Umgebungen?
   - Warten wir mit der Bereitstellung in der Produktion bis zu einem geplanten Release-Zeitpunkt?
   - Nehmen wir regelmäßig Updates an der Konsole der Produktionsumgebung vor?
   - Gibt es manuelle Bereitstellungsschritte, die in der Konsole der Produktionsumgebung ausgeführt werden müssen, um die Bereitstellung abzuschließen?
   - Hat mehr als eine Person Schreibzugriff auf die Produktionsumgebung?
   - Haben mehr als fünf Personen Schreibzugriff auf die Produktionsumgebung?

4. Teilen Sie Ihre Punktzahl durch 2. Dies ist der CI/CD-Score Ihrer Pipeline.
5. Vergleichen Sie das Niveau Ihrer Pipeline. CI/CD score to the following table to determine your pipeline's CI/CD

CI/CD-Punktzahl	CI/CD-Stufe
9.5 oder höher	Vollständig CI/CD
8—9	Hauptsächlich CI/CD
5—7	Neutral
Unter 5	Nicht CI/CD

Wenn Sie weniger als 8 Punkte erzielt haben, empfehlen wir Ihnen, sich ein Ziel zu setzen, um schrittweise zum nächsten Level zu gelangen. Wenn dieses Ziel erreicht ist, sollten die Produktbeteiligten prüfen, ob und wann ein neues Ziel festgelegt werden sollte. Mit dieser Maßnahme

soll nicht unbedingt eine Änderung Ihrer Pipeline befürwortet werden, sondern es soll vielmehr ein Bewusstsein dafür geschaffen werden, wie ein vollständiger CI/CD-Implementierungsprozess aussieht und wo sich Ihre Pipelines derzeit auf diesem Spektrum befinden.

Bewährte Methoden für CI/CD-Pipelines

Im Folgenden finden Sie bewährte Methoden für vollständige CI/CD-Pipelines:

- Sicherung der Produktionsumgebung — Da mit IaC praktisch alles Notwendige für die Konto- und Umgebungswartung erledigt werden kann, ist es wichtig, alle Anstrengungen zu unternehmen, um die Produktionsumgebung zu schützen, indem der Konsolen- und programmgesteuerte Zugriff eingeschränkt wird. Wir empfehlen, den Zugriff auf nur wenige oder gar keine Benutzer zu beschränken. Wenn Sie IaC über bereitstellen AWS CloudFormation, benötigt der Benutzer eingeschränkte Berechtigungen. Die meisten Berechtigungen werden dem CloudFormation Dienst über eine Servicerolle zugewiesen. Weitere Informationen finden Sie in der CloudFormation Dokumentation unter Servicerolle und unter Implementieren von Richtlinien für Berechtigungen mit den geringsten Rechten für. AWS CloudFormation
- Erstellen Sie separate Konten für jede Umgebung — Indem Sie jeder Umgebung ein eigenes Konto zuweisen, können Sie den Bereitstellungsprozess vereinfachen und detaillierte Zugriffskontrollen auf Kontoebene einrichten. Wenn mehrere Umgebungen Ressourcen gemeinsam nutzen, verringert dies die Integrität der Umgebung als isolierte Einheit. Es ist am besten, Umgebungen zu synchronisieren und voneinander zu unterscheiden. Dies ist für die Produktionsumgebung noch wichtiger, da alles, was in diesem Konto enthalten ist, als Produktionsressource behandelt werden sollte.
- Beschränken Sie persönlich identifizierbare Informationen (PII) auf die Produktionsumgebung — Sowohl aus Sicherheitsgründen als auch zum Schutz vor Haftungsrisiken sollten Sie personenbezogene Daten so weit wie möglich schützen. Verwenden Sie nach Möglichkeit in niedrigeren Umgebungen anonymisierte Daten oder Beispieldaten, anstatt potenziell sensible Daten aus der Produktionsumgebung zu kopieren.
- Code in Repositorys überprüfen — Ein vollständiger CI/CD-Prozess reduziert die Zugangspunkte für eine Pipeline auf einen einzigen Punkt, und dieser zentrale Punkt sollte gesichert werden. Aus diesem Grund wird empfohlen, dass Sie mehrere Code-Reviews durchführen lassen, bevor Sie Feature-Branches in den Hauptzweig zusammenführen. Diese Code-Reviews können von jedem qualifizierten Teammitglied durchgeführt werden, aber mindestens ein erfahrenes Mitglied sollte sie überprüfen. Der Code sollte vom Prüfer gründlich getestet werden. Schließlich besteht der beste Weg, Probleme in einer Pipeline zu beheben, darin, zu vermeiden, dass sie in die Pipeline eingeführt werden. Außerdem ist es wichtig, alle Kommentare aller Prüfer vor der Zusammenführung zu klären. Diese Lösung könnte lediglich eine Erklärung dafür sein, warum keine Änderungen erforderlich sind, aber die Berücksichtigung aller Kommentare ist eine wichtige zusätzliche Überprüfung, um zu verhindern, dass Probleme in die Pipeline kommen.

- Kleine und häufige Zusammenführungen vornehmen — Um die Vorteile der kontinuierlichen Integration voll auszuschöpfen, ist es eine gute Idee, auch lokale Änderungen kontinuierlich in die Pipeline zu integrieren. Schließlich ist es für die Entwicklungsumgebungen viel vorteilhafter, synchron zu bleiben, wenn auch die lokalen Umgebungen mit ihnen Schritt halten.

Weitere bewährte Methoden für CI/CD-Pipelines finden Sie unter Zusammenfassung der bewährten Methoden in Practicing Continuous Integration and Continuous Delivery am. AWS

Häufig gestellte Fragen

Welche wichtigen Indikatoren deuten darauf hin, dass mein Bereitstellungsprozess nicht vollständig CI/CD ist?

Der häufigste Indikator ist, wenn es mehrere Repository-Zweige gibt, die separate Umgebungen in einer Pipeline repräsentieren. Repositorys in einem vollständigen CI/CD-Prozess verwenden einen Trunk-basierten Workflow, bei dem ein Zweig als zentrale Informationsquelle für die Bereitstellungen dieses Repositorys fungiert. Weitere Informationen finden Sie unter Trunk-basierter Ansatz. Zu den weiteren Indikatoren gehören manuelle Bereitstellungsschritte, bei denen es sich nicht um einfache Entscheidungen handelt, ob sie losgehen oder nicht, die Verwendung von Hotfixes und geplante Releases.

Was ist, wenn ich einen vollständigen CI/CD-Prozess nutzen möchte, aber dennoch Releases bestimmter Funktionen für bestimmte Zeitpunkte planen möchte?

Dies geschieht normalerweise mit Feature-Flags. In diesem Prozess werden Bereitstellungen immer noch kontinuierlich vorgenommen, aber bestimmte Funktionen werden durch bedingte Schließungen im Code ausgeblendet, bis es Zeit ist, sie zu veröffentlichen.

Was ist, wenn einige Schritte in meinem Bereitstellungsprozess nicht automatisiert werden können?

Eines der Ziele einer vollständigen CI/CD-Pipeline besteht darin, den Bedarf an manuellen Prozessen zu minimieren, aber es gibt sicherlich potenzielle Anwendungsfälle, in denen manuelle Prozesse erforderlich sein können. Tatsächlich können schreibgeschützte Prozesse, wie z. B. das Abrufen von Anwendungsprotokollen, häufig in Produktionsumgebungen mit minimalem Risiko durchgeführt werden. Es wird jedoch dringend empfohlen, manuelle Schreibaktionen in der Produktion als absoluten letzten Ausweg zu behandeln.

Was ist, wenn mein technisches Personal mit älteren Workflows besser vertraut ist als mit einem vollständigen CI/CD-Prozess?

Es ist üblich, dass technisches Personal großen Änderungen widersteht, insbesondere wenn etwas, das früher als bewährte Methode galt, durch etwas Neueres ersetzt wird. Die Technologie entwickelt sich schnell und es werden ständig Verbesserungen entdeckt. Ein gewisses Maß an Skepsis ist zwar eine gute Eigenschaft eines technischen Personals, aber es ist genauso wichtig, dass es offen für Veränderungen ist. Gehen Sie nicht zu schnell mit skeptischen Mitarbeitern um, da sie Änderungen am System verwalten müssen, bevor sie implementiert werden. Der Schlüssel liegt darin, zu verhindern, dass Skeptiker für immer statisch bleiben.

Was ist, wenn sich meine Umgebungen in mehreren Konten befinden? Kann ich trotzdem einen vollständigen CI/CD-Prozess verwenden?

Ja, es wird sogar empfohlen, für jede Umgebung ein separates Konto zu verwenden. Weitere Informationen zu einer Pipeline, die Stufen in verschiedenen Konten aktiviert, finden Sie unter Erstellen einer Pipeline in CodePipeline , die Ressourcen aus anderen Konten verwendet AWS-Konto.

Nächste Schritte

Verwenden Sie diesen Lackmustest für CI/CD-Pipelines Abschnitt, um die DevOps Prozesse in Ihrer Organisation zu bewerten. Stellen Sie fest, ob es sich bei Ihren Prozessen um vollständige CI/CD. If they aren't, decide whether they need improvement to take full advantage of the benefits of CI/CD Implementierungen handelt.

Woher wissen Sie, wann Sie fertig sind? Nun, die Antwort ist, dass viele Organisationen nie wirklich fertig werden. Sie halten irgendwo auf dem Weg an, an einer Stelle, die für ihren Anwendungsfall geeignet ist. Eine vollständige CI/CD pipeline is the best-case scenario, it depends heavily on the organizational situation and the stakeholders behind the decision. Stakeholders must decide which stage of the CI/CD Implementierung eignet sich jedoch am besten für ihren Anwendungsfall und dafür, wie der Fortschritt am besten den nächsten Phasen zugeordnet werden kann.

Weitere Informationen zum Entwerfen und Erstellen von CI/CD-Pipelines finden Sie unter. Ressourcen

Ressourcen

AWS Dokumentation und Referenzen

- AWS Deployment Pipeline Reference Architecture
- Was ist Continuous Delivery?
- Praktische Umsetzung von Continuous Integration und Continuous Delivery AWS (AWS Whitepaper)
- Richten Sie eine CI/CD-Pipeline ein AWS(AWS praktisches Tutorial)
- Erstellen Sie eine Pipeline AWS-Regionen , die nicht unterstützt wird (Prescriptive Guidance AWS CodePipeline)AWS
- Referenzarchitektur und Referenzimplementierungen für Bereitstellungspipelines (Blogbeitrag)AWS

Dienste und Tools

- Der CI/CD Litmus Test
- AWS Cloud Development Kit (AWS CDK)
- AWS CloudFormation
- AWS CodePipeline

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	—	25. August 2023

AWS Glossar zu präskriptiven Leitlinien

Die folgenden Begriffe werden häufig in Strategien, Leitfäden und Mustern verwendet, die von AWS Prescriptive Guidance bereitgestellt werden. 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 ein 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 Transaktionen von verbindenden Anwendungen verarbeitet, 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 den erfolgreichen Umstieg 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, die als bösartige Bots bezeichnet werden, 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 für die AWS 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 or Bitbucket Cloud. Jede Version des Codes wird als Zweig bezeichnet. 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. AWS Panorama Bietet beispielsweise Geräte an, die CV zu lokalen Kameranetzwerken hinzufügen, und Amazon SageMaker AI stellt Bildverarbeitungsalgorithmen für CV bereit.

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 is commonly described as a pipeline. CI/CDkann 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.

Bereitstellung

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 der Wertströme 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 Sicherheitsthemen 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 so 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, 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 einer Organisation angehören.

DURCHEINANDER

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

Weitere Informationen finden Sie unter 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 Erstellen eines Pfads für eine Organisation.

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. Weitere Informationen finden Sie unter Datenpersistenz in Microservices aktivieren.

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 mit der Steuerung konform ist, 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).

LAPPEN

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 der Wiederherstellungszeit.

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 Management Console 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 EC2 Amazon-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, wohingegen 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

Mal 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.