Vorbereitung auf Infrastrukturereignisse

Leitfaden und bewährte Methoden für AWS

Juli 2017

Hinweise

Dieses Dokument wird nur zu Informationszwecken zur Verfügung gestellt. Es stellt das aktuelle Produktangebot und die Verfahren von AWS zum Ausstellungsdatum dieses Dokuments dar. Änderungen vorbehalten. Kunden sind für ihre eigene unabhängige Einschätzung der Informationen in diesem Dokument und jedwede Nutzung der AWS-Services verantwortlich. Jeder Service wird „wie besehen“ ohne Gewähr und ohne Garantie jeglicher Art, weder ausdrücklich noch impliziert, bereitgestellt. Dieses Dokument gibt keine Garantien, Gewährleistungen, vertraglichen Verpflichtungen, Bedingungen oder Zusicherungen von AWS, seinen Partnern, Zulieferern oder Lizenzgebern. Die Verantwortung und Haftung von AWS gegenüber seinen Kunden werden durch AWS-Vereinbarungen geregelt. Dieses Dokument ist weder ganz noch teilweise Teil der Vereinbarungen von AWS mit seinen Kunden und ändert diese Vereinbarungen auch nicht.

Inhalt

Kurzbeschreibung

In diesem Whitepaper werden Richtlinien und bewährte Methoden für Kunden mit Produktions-Workloads auf Amazon Web Services (AWS) beschrieben, die cloudbasierte Anwendungen so optimieren und bereitstellen möchten, dass sie geplante Skalierungsereignisse, z. B. Produkteinführungen oder saisonale Datenverkehrsspitzen, problemlos und dynamisch bewältigen können. Es werden allgemeine Designgrundsätze sowie spezifische bewährte Methoden vorgestellt. Außerdem werden verschiedene konzeptionelle Aspekte der Infrastrukturereignisplanung näher betrachtet. Abschließend sprechen wir über Einflussfaktoren auf die Betriebsbereitschaft, damit verbundene Verfahrensweisen und dem Ereignis nachgelagerte Aktivitäten.

Einleitung

Bei der Vorbereitung auf Infrastrukturereignisse geht es darum, auf erwartete und wichtige Ereignisse, die Auswirkungen auf Ihr Unternehmen haben, vorbereitet zu sein und dementsprechend zu planen. Dabei handelt es sich um Ereignisse, bei denen es unbedingt erforderlich ist, dass der Webservice des Unternehmens zuverlässig, schnell und hochgradig fehlertolerant funktioniert, und zwar unter allen Bedingungen und bei Änderungen an den Datenverkehrsmustern. Solche Ereignisse sind zum Beispiel die Expansion in neue Regionen, die Einführung neuer Produkte oder Funktionen, saisonbedingte Ereignisse oder bedeutende geschäftliche Ankündigungen oder Marketing-Events.

Ein Infrastrukturereignis, auf das Sie nicht richtig vorbereitet sind, kann negative Auswirkungen auf Ihren geschäftlichen Ruf, die Funktionsfähigkeit Ihres Unternehmens oder Ihr finanzielles Ergebnis haben. Infrastrukturereignisbedingte Ausfälle können sich in verschiedenen Formen manifestieren: als unvorhergesehene Serviceausfälle, Leistungseinbruch aufgrund Überlastung, Netzwerklatenz, Erreichen der Speicherkapazitäts- oder Systemgrenzen, z. B. der API-Aufrufrate, Erreichen einer endlichen Menge an verfügbaren IP-Adressen, als mangelndes Wissen über das Verhalten von Komponenten eines Anwendungs-Stacks aufgrund unzureichender Überwachung, unvorhergesehene Abhängigkeiten von einem Drittanbieter oder einer Komponente, die nicht für eine Skalierung ausgelegt ist, oder als eine andere unvorhergesehene Fehlerbedingung.

Um das Risiko unerwarteter Ausfälle während eines wichtigen Ereignisses zu minimieren, sollten Unternehmen Zeit und Ressourcen in die Planung und Vorbereitung investieren, Mitarbeiter schulen und relevante Prozesse entwickeln und dokumentieren. Wie groß letztlich der Aufwand ist, der in die Infrastrukturereignisplanung für eine bestimmte cloudfähige Anwendung oder mehrere Anwendungen fließt, hängt von der Komplexität und globalen Reichweite des Systems ab. Die in diesem Whitepaper vorgestellten Designprinzipien und bewährten Methoden gelten unabhängig vom Umfang oder der Komplexität der Cloudpräsenz des Unternehmens.

Mit Amazon Web Services (AWS) kann Ihr Unternehmen seine Infrastruktur in Vorbereitung auf ein geplantes Skalierungsereignis auf „Pay-as-you-go“-Basis dynamisch und anpassbar skalieren. Mit einer umfassenden Palette von

elastischen und programmierbaren Produkten und Services bietet Amazon Ihrem Unternehmen Zugriff auf dieselbe hochgradig sichere, zuverlässige und schnelle Infrastruktur, auf der Amazon sein eigenes globales Netzwerk betreibt, und ermöglicht es Ihnen, sich in kürzester Zeit an neue geschäftliche Gegebenheiten anzupassen.

Die in diesem Whitepaper beschriebenen bewährten Methoden und Designprinzipien helfen Ihnen, für ein Infrastrukturereignis zu planen und diese Planung umzusetzen. Sie erfahren außerdem, wie Sie mit den AWS-Produkten Ihre Anwendungen nach Bedarf vertikal und horizontal skalieren können.

Planung und Vorbereitung auf ein Infrastrukturereignis

In diesem Abschnitt erfahren Sie, was ein geplantes Infrastrukturereignis ausmacht und welche Aktivitäten in der Regel mit einem derartigen Ereignis verbunden sind.

Was ist ein geplantes Infrastrukturereignis?

Ein *geplantes Infrastrukturereignis* ist ein aus geschäftlichen Gründen erwartetes, geplantes und zeitlich begrenztes Ereignis, während dem es unbedingt erforderlich ist, dass ein Webservice schnell, skalierbar und fehlertolerant funktioniert. Der Grund hierfür können Marketing-Kampagnen, Nachrichtenereignisse aufgrund der geschäftlichen Tätigkeit des Unternehmens, Produkteinführungen, die Expansion in neue Regionen oder ähnliche Aktivitäten sein, welche die webbasierten Anwendungen und die zugrunde liegende Infrastruktur des Unternehmens mit zusätzlichem Datenverkehr belasten.

Was geschieht während eines geplanten Infrastrukturereignisses?

Bei den meisten geplanten Infrastrukturereignissen geht es im Wesentlichen darum, zusätzliche Kapazität hinzufügen zu können, damit die Web-Infrastruktur den Anstieg beim Datenverkehr bewältigen kann. In einer herkömmlichen lokalen Umgebung mit physischen Datenverarbeitungs-,

Speicher- und Netzwerkressourcen müsste die IT-Abteilung zusätzliche Kapazität basierend auf fundierten Schätzungen der theoretischen maximalen Spitzenlast hinzufügen. Diese Methode birgt das Risiko einer unzureichenden Kapazitätsbereitstellung, die dem Unternehmen im schlimmsten Fall aufgrund von überlasteten Webservern, langsamen Reaktionszeiten und anderen Laufzeitfehlern geschäftliche Verluste einbringt.

Innerhalb der AWS Cloud ist die Infrastruktur programmierbar und elastisch. Das bedeutet, dass sie als Reaktion auf aktuelle Anforderungen in kürzester Zeit bereitgestellt werden kann. Außerdem kann sie so konfiguriert werden, dass sie auf automatisierte, intelligente und dynamische Weise auf Systemmetriken reagiert, zum Beispiel auf wachsende oder schrumpfende Ressourcen wie Web-Server-Cluster, bereitgestellten Durchsatz, die Speicherkapazität, die verfügbaren Rechenkerne, die Anzahl der Streaming-Shards usw.

Darüber hinaus sind viele AWS-Produkte vollständig verwaltet. Dazu gehören unter anderem Speicher-, Datenbank-, Analyse-, Anwendungs- und Bereitstellungs-Services. AWS-Kunden haben daher den Vorteil, sich keine Gedanken über die aufwendige Konfiguration dieser Services für ein Ereignis mit hoher Datenverkehrslast machen zu müssen. Die vollständig verwalteten Services (Managed Services) von AWS wurden für Skalierbarkeit und hohe Verfügbarkeit entwickelt.

Vorbereitend auf ein geplantes Infrastrukturereignis führen AWS-Kunden normalerweise einen Systemcheck durch. Sie bewerten die Architektur und Betriebsbereitschaft ihrer Anwendung und stellen dabei auch die Skalierbarkeit und Fehlertoleranz auf den Prüfstand. Datenverkehrsschätzungen werden herangezogen und mit der Performance im normalen Geschäftsbetrieb verglichen. Zudem werden Kapazitätsmetriken festgelegt und die zusätzlich benötigte Kapazität wird geschätzt. Alle möglichen Engpässe sowie Upstream- und Downstream-Abhängigkeiten von Drittanbietern werden identifiziert und fließen in die Betrachtung ein. Handelt es sich bei dem geplanten Ereignis um die Expansion in eine neue Region oder die Ansprache neuer Zielgruppen, werden auch geografische Aspekte berücksichtigt. Expansionen in zusätzliche AWS-Regionen und Availability Zones finden vor dem geplanten Ereignis statt. Auch die kundenspezifischen dynamischen Systemeinstellungen von AWS, wie z. B. Auto Scaling, Load Balancing, Geo-Routing, hohe Verfügbarkeit und Failover-Maßnahmen, werden überprüft, um sicherzustellen, dass sie so konfiguriert sind, dass sie den erwarteten Datenverkehrsanstieg und die

höheren Transaktionsraten bewältigen können. Statische Einstellungen wie AWS-Ressourcenlimits und der Standort der Ursprungs-Server des Netzwerks zur Bereitstellung von Inhalten (Content Delivery Network, CDN) werden ebenfalls begutachtet und bei Bedarf angepasst.

Dasselbe gilt für die Überwachungs- und Benachrichtigungsmechanismen, denn diese sollen während des Ereignisses in Echtzeit Informationen liefern und nach Abschluss des geplanten Ereignisses eine Post-mortem-Analyse ermöglichen.

Während des geplanten Ereignisses können AWS-Kunden für den Fall, dass Fehler behoben werden müssen oder Echtzeit-Support erforderlich ist, z. B. wenn ein Server ausfällt, Support-Vorgänge bei AWS erstellen. Kunden mit AWS Enterprise Support-Plan können zusätzlich jederzeit mit Support-Technikern in Kontakt treten und Vorgänge mit kritischem Schweregrad erstellen, wenn eine besonders schnelle Reaktion erforderlich ist.

Die AWS-Ressourcen sind so konzipiert, dass sie am Ende eines Ereignisses dem Datenverkehrsaufkommen entsprechend automatisch auf das angemessene Maß zurückskalieren, oder eben noch weiter hochskalieren, wenn das Ereignis es erfordert.

Designprinzipien

Die Vorbereitung auf geplante Ereignisse beginnt mit einem guten Design, und zwar schon zu Anfang jeder Implementierung eines cloudbasierten Anwendungs-Stacks oder einer Workload.

Separate Workloads

Gutes Design ist unerlässlich, damit die Workloads eines geplanten Ereignisses bei normalem und erhöhtem Datenverkehr effizient verwaltet werden können. Achten Sie von Beginn an darauf, separate und unabhängige funktionelle Ressourcengruppen zu erstellen, die auf eine bestimmte Geschäftsanwendung oder ein Produkt ausgelegt sind. In diesem Abschnitt wird auf die verschiedenen Aspekte dieses Designziels eingegangen.

Tagging

Tags dienen dem Benennen und Organisieren von Ressourcen. Sie sind ein wesentlicher Bestandteil der Verwaltung von Infrastrukturressourcen während

eines geplanten Infrastrukturereignisses. In AWS sind Tags vom Kunden verwaltete Labels mit einem Schlüsselwert, die auf eine einzelne verwaltete Ressource angewendet werden, z. B. auf einen Load Balancer oder eine Amazon Elastic Compute Cloud (EC2) Instance. Gut definierte Tags, die den AWS-Ressourcen zugeordnet sind, erleichtern es Ihnen herauszufinden, welche Ressourcen in Ihrer Gesamtinfrastruktur die Workload des geplanten Ereignisses ausmachen. Mithilfe dieser Informationen können Sie dann die Bereitschaft Ihrer Infrastruktur analysieren. Tags können auch der korrekten Kostenzuordnung dienen.

Mit Tags lassen sich zum Beispiel EC2 Instances, Amazon Machine Image(AMI)-Images, Load Balancer, Sicherheitsgruppen, Amazon Relational Database Service(RDS)-Ressourcen, Amazon Virtual Private Cloud(VPC)-Ressourcen, Amazon Route 53-Zustandsprüfungen und Amazon Simple Storage Service(S3)-Buckets organisieren.

Weitere Informationen zu effektiven Tagging-Strategien finden Sie in AWS-Tagging-Strategien.[1]

Weitere Beispiele für das Erstellen und Verwalten von Tags in Ressourcengruppen finden Sie unter Ressourcengruppen und Tagging für AWS.[2]

Lose Verkoppelung

Wenn Sie für eine Cloud-Architektur entwickeln, sollte jede Komponente des Anwendungs-Stacks so weit wie möglich voneinander unabhängig sein. Dadurch werden cloudbasierte Workloads elastisch und skalierbar.

Sie können die Abhängigkeiten zwischen den Komponenten in einem cloudbasierten Anwendungs-Stack reduzieren, indem Sie jede Komponente als eine Blackbox mit genau definierten Schnittstellen für Ein- und Ausgaben (z. B. RESTful-APIs) entwickeln. Sind die Komponenten keine Anwendungen, sondern Services, die zusammen eine Anwendung ergeben, nennt man das *Microservices-Architektur*. Für die Kommunikation und Koordination zwischen Anwendungskomponenten können Sie ereignisgesteuerte Benachrichtigungsmechanismen wie AWS-Nachrichtenwarteschlangen verwenden, die Benachrichtigungen zwischen den Komponenten übergeben, wie in Abbildung 1 dargestellt.

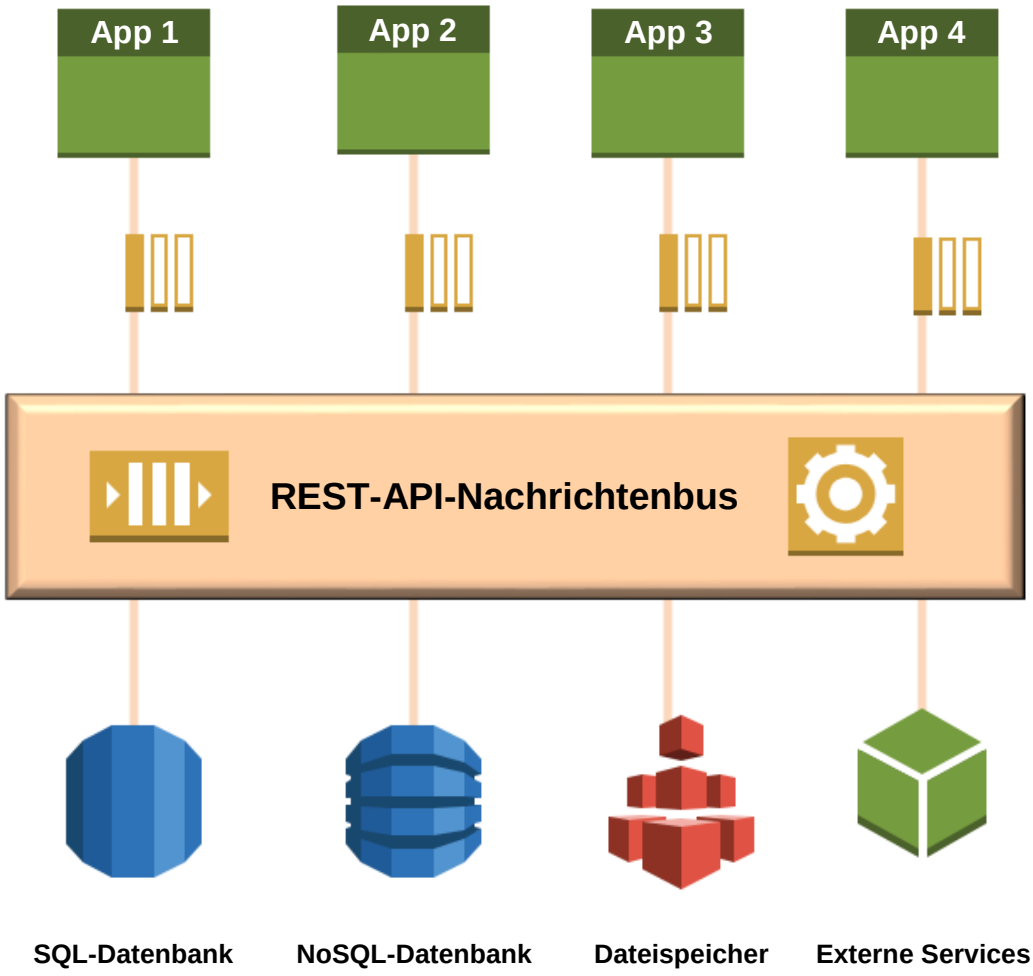

Abbildung 1. Lose Verkoppelung mithilfe von RESTful-Schnittstellen und Nachrichtenwarteschlangen

Wenn Sie Mechanismen wie diese verwenden, ist es viel weniger wahrscheinlich, dass eine Änderung oder ein Fehler in einer Komponente sich auf andere Komponenten auswirken kann. Angenommen, ein Server in einem mehrschichtigen Anwendungs-Stack reagiert nicht mehr. Lose verkoppelte Anwendungen könnten die entsprechende Schicht einfach umgehen oder im Degraded-Modus auf alternative Transaktionswege wechseln.

Auch die asynchrone Integration lässt sich mit lose verkoppelten Anwendungskomponenten mit zwischengeschalteten Nachrichtenwarteschlangen einfacher realisieren. Da die Komponenten einer Anwendung keine direkte P2P-Kommunikation nutzen, sondern stattdessen einen zwischengeschalteten und persistenten Messaging-Layer verwenden (z. B.

eine Amazon Simple Queue Service(SQS)-Warteschlange oder einen Streaming-Daten-Mechanismus wie Amazon Kinesis Streams), kommen sie mit einem plötzlichen Aktivitätsanstieg in einer Komponente gut zurecht, während die nachgelagerten Komponenten die Eingangswarteschlange verarbeiten. Fällt eine Komponente aus, bleiben die Nachrichten in den Warteschlangen oder Streams erhalten, bis die ausgefallene Komponente wieder funktioniert.

Weitere Informationen zu Nachrichtenwarteschlangen und den Benachrichtigungs-Services von AWS finden Sie in der Dokumentation zu Amazon Simple Queue Service.[3]

Services, nicht Server

Mit Managed Services und Service-Endpunkten müssen Sie sich über Sicherheit und Zugriff, Backup und Wiederherstellung, Patch-Management oder Änderungskontrolle, die Einrichtung von Überwachung und Berichterstellung oder die vielen anderen Details im Zusammenhang mit herkömmlichen Systemverwaltungsvorgängen keine Gedanken mehr machen. Diese Cloud-Ressourcen können vorab in Konfigurationen mit mehreren Availability Zones (oder in manchen Fällen mehreren Regionen) bereitgestellt werden, um für hohe Verfügbarkeit und Ausfallsicherheit zu sorgen. Sie können nach oben oder unten skaliert werden, oftmals sogar ohne Ausfallzeit. Und Sie können im Betrieb entweder über die AWS Management Console oder API/CLI-Aufrufe konfiguriert werden.

Mit Managed Services und Service-Endpunkten lassen sich Kunden-Anwendungs-Stacks um Funktionen wie relationale und NoSQL-Datenbanken, Data Warehousing, Ereignisbenachrichtigung, Objekt- und Dateispeicher, Echtzeit-Streaming, Big Data-Analysen, maschinelles Lernen, Suche, Codeumwandlung und vieles mehr erweitern. Ein Endpunkt ist eine URL, die einem AWS-Produkt als Einstiegspunkt dient. So ist zum Beispiel https://dynamodb.us-west-2.amazonaws.com ein Einstiegspunkt für den Amazon DynamoDB-Service.

Managed Services und ihre Service-Endpunkte ermöglichen es Ihnen, produktionsbereite Ressourcen als Teil Ihrer Lösung zu verwenden, um die während eines geplanten Infrastrukturereignisses anfallenden höheren Datenlasten und Transaktionsraten zu verarbeiten und die Reichweite zu vergrößern. Sie müssen keine eigenen Server bereitstellen und verwalten, die dieselben Funktionen bieten wie die Managed Services.

Weitere Informationen zu AWS-Service-Endpunkten finden Sie unter AWS-Regionen und -Endpunkte.[4] Siehe auch Amazon EMR,[5] Amazon RDS[6] und Amazon ECS[7] für Beispiele von Managed Services mit Endpunkten.

Architekturen ohne Server

Eine weitere Strategie zur effizienten Reaktion auf sich dynamisch ändernde Arbeitslasten während eines geplanten Infrastrukturereignisses bietet die Nutzung von AWS Lambda. Lambda ist eine ereignisgesteuerte Serverless-Computing-Plattform. Der dynamisch aufgerufene Service führt beim Eintreten bestimmter Ereignisse (über Benachrichtigungen) Node.js-, Python- oder Java-Code aus und verwaltet die in dem Code spezifizierten Datenverarbeitungsressourcen automatisch. Lambda erfordert keine Vorabbereitstellung von Amazon EC2-Datenverarbeitungsressourcen. Der Amazon Simple Notification Service (Amazon SNS) kann so konfiguriert werden, dass er Lambda-Funktionen auslöst. Weitere Informationen zu Amazon SNS finden Sie unter Amazon Push Notification Service.[8]

Die Serverless-Funktionen von Lambda können Code ausführen, der auf andere AWS-Produkte zugreift oder diese aufruft. Das können Datenbankoperationen, Datentransformationen, Objekt- oder Dateiabrufvorgänge und sogar Skalierungsvorgänge als Reaktion auf externe Ereignisse oder interne Systemlastmetriken sein. AWS Lambda kann auch neue Benachrichtigungen oder Ereignisse generieren und andere Lambda-Funktionen starten.

AWS Lambda ermöglicht die Feinsteuerung von Skalierungsvorgängen während eines geplanten Infrastrukturereignisses. So kann Lambda verwendet werden, um Auto Scaling erweiterte Aktionen ausführen zu lassen. Zum Beispiel können Drittanbietersysteme benachrichtigt werden, die ebenfalls skaliert werden müssen, oder neu bereitgestellten Instances werden zusätzliche Netzwerkschnittstellen hinzugefügt. Beispiele dafür, wie Sie mit Lambda Skalierungsvorgänge anpassen können, finden Sie unter Using AWS Lambda with Auto Scaling Lifecycle Hooks[9].

Weitere Informationen zu AWS Lambda finden Sie unter What is AWS Lambda?[10]

Automatisierung

Auto Scaling

Auto Scaling ist ein wichtiges Element bei der Planung eines Infrastrukturereignisses. Die Fähigkeit, die Kapazität einer Anwendung gemäß vordefinierter Bedingungen automatisch nach oben oder unten zu skalieren, ermöglicht es, die Verfügbarkeit der Anwendung auch bei Schwankungen in den Datenverkehrsmustern und der Datenverkehrslast aufrechtzuerhalten, die während eines geplanten Infrastrukturereignisses auftreten.

AWS stellt dafür vielen seiner Ressourcen, einschließlich EC2 Instances, Datenbankkapazität, Container usw., die Auto Scaling-Funktion bereit.

Mit Auto Scaling können Instances-Gruppen skaliert werden. Zum Beispiel lassen sich mehrere Server, die eine cloudbasierte Anwendung unterstützen, basierend auf bestimmten Bedingungen automatisch skalieren. Auto Scaling kann auch verwendet werden, um eine feste Anzahl von Instances aufrechtzuerhalten, selbst wenn eine Instance fehlerhaft ist. Diese automatische Skalierung und die Aufrechterhaltung der Anzahl von Instances ist die Kernfunktionalität des Auto Scaling-Service.

Auto Scaling sorgt für eine gleichbleibende Anzahl von Instances, indem die Instances in der Gruppe regelmäßig einer Zustandsprüfung unterzogen werden. Eine fehlerhafte Instance wird von der Gruppe beendet und als Ersatz wird eine neue Instance gestartet.

Auto Scaling-Richtlinien ermöglichen es, die Anzahl ausgeführter EC2 Instances in einer Servergruppe als Reaktion auf veränderte Bedingungen automatisch zu erhöhen oder zu verringern. Wenn eine Skalierungsrichtlinie in Kraft ist, passt die Auto Scaling-Gruppe die benötigte Kapazität automatisch an und startet oder beendet Instances nach Bedarf, entweder dynamisch oder alternativ nach einem Zeitplan, falls die Zu- und Abnahme des Datenverkehrs zu vorhersehbaren Zeiten erfolgt.

Neustarts und Wiederherstellung

Für jedes geplante Infrastrukturereignis gilt, dass es definierte Verfahren und eine Automatisierung für den Umgang mit kompromittierten Instances oder Servern geben sollte, um diese im laufenden Betrieb wiederherstellen oder neu starten zu können.

EC2 Instances können so eingerichtet werden, dass sie automatisch wiederhergestellt werden, wenn eine Systemzustandsprüfung der zugrunde liegenden Hardware einen Fehler ausgibt. Die Instance wird neu gestartet (wenn nötig auf neuer Hardware), behält aber Instance-ID, IP-Adresse, Elastic IP-Adressen, zugewiesene Amazon Elastic Block Store(EBS)-Volumes und andere Konfigurationsdetails bei. Weitere Informationen über die automatische Wiederherstellung von EC2 Instances finden Sie unter Automatische Wiederherstellung von Amazon EC2.[11]

Konfigurationsmanagement/Orchestrierung

Ein weiteres wichtiges Element einer robusten, zuverlässigen und flexiblen geplanten Infrastrukturereignisstrategie sind Konfigurationsmanagement- und Orchestrierungs-Tools für das Zustandsmanagement einzelner Ressourcen und die Anwendungs-Stack-Bereitstellung.

Konfigurationsmanagement-Tools übernehmen in der Regel die Bereitstellung und Konfiguration von Server-Instances, Load Balancers, Auto Scaling, die Bereitstellung einzelner Anwendungen und deren Statusüberwachung. Sie bieten außerdem die Möglichkeit zur Integration weiterer Services wie Datenbanken, Speicher-Volumes und Caching-Ebenen.

Orchestrierungs-Tools stehen eine Abstraktionsebene über den Konfigurationsmanagement-Tools. Sie ermöglichen es, Beziehungen zwischen den verschiedenen Ressourcen herzustellen. Kunden können dadurch mehrere Ressourcen als vereinheitlichte Cloud-Anwendungsinfrastruktur bereitstellen und verwalten, ohne sich Gedanken über Ressourcenabhängigkeiten machen zu müssen.

Die Tools definieren und beschreiben die einzelnen Ressourcen und deren Abhängigkeiten untereinander als Code, der einer Versionskontrolle unterstellt werden kann. Dies wiederum ermöglicht es, ein Rollback auf frühere Versionen durchzuführen, oder im Rahmen von Test und Entwicklung neue Codevarianten auszuprobieren. So ist es auch möglich, für ein Infrastrukturereignis optimierte Orchestrierungen und Konfigurationen zu definieren und nach dem Ereignis ein Rollback auf die Standardkonfiguration durchzuführen.

Amazon Web Services empfiehlt die folgenden Tools für Bereitstellungen und Orchestrierungen von Hardware als Code:

- **AWS Config mit Config Rules** oder ein AWS Config-Partner für eine detaillierte, visuelle und durchsuchbare Bestandsaufnahme der AWS-Ressourcen mit Konfigurationsverlauf und konformer Ressourcenkonfiguration.
- **AWS CloudFormation** oder Drittanbieter-Tools zur Orchestrierung und Verwaltung der Bereitstellung, Aktualisierung und Beendigung von AWS-Ressourcen.
- **AWS OpsWorks Elastic Beanstalk** oder Serverkonfigurationsmanagement-Tools von Drittanbietern zur Verwaltung von Betriebssystem- und Anwendungskonfigurationsänderungen.

Weitere Informationen zu den verschiedenen Möglichkeiten, Hardware als Code zu verwalten, finden Sie unter Infrastrukturkonfigurationsmanagement.[12]

Vielfalt/Ausfallsicherheit

Entfernen von „Single Points of Failure“ und Engpässen

Bei der Planung eines Infrastrukturereignisses sollten Sie Ihre Anwendungs-Stacks auf einzelne Fehlerquellen (Single Points of Failure (SPOF)) oder Performance-Engpässe untersuchen. Gibt es zum Beispiel eine einzelne Instance eines Servers, Daten-Volumes, einer Datenbank, eines NAT-Gateways oder Load Balancers, bei deren Ausfall die gesamte Anwendung oder erhebliche Teile davon nicht mehr funktionieren?

Und zweitens: Wenn die cloudbasierte Anwendung abhängig vom Datenverkehr oder Transaktionsvolumen nach oben skaliert, gibt es einen Teil der Infrastruktur, der an eine physische Grenze oder auf eine Einschränkung stößt, wie die Netzwerkbandbreite oder CPU-Rechenzyklen, sobald immer mehr Daten über den Datenflusspfad geleitet werden?

Diese Risiken können auf unterschiedliche Weise abgemildert werden.

Ausfallsicheres Design

Wie bereits erwähnt, bieten lose Verkoppelung und Nachrichtenwarteschlangen mit RESTful-Schnittstellen einen gangbaren Weg, um Ausfallsicherheit für einzelne Ressourcen zu erreichen oder Fluktuationen im Datenverkehrsaufkommen oder Transaktionsvolumen abzufangen. Eine weitere

Regel ausfallsicheren Designs ist es, Anwendungskomponenten so zustandslos wie möglich zu konfigurieren.

Zustandslose Anwendungen erfordern kein Wissen über frühere Transaktionen und sind nur lose von anderen Anwendungskomponenten abhängig. Sie speichern keine Sitzungsinformationen. Eine zustandslose Anwendung kann als Mitglied eines Pools oder Clusters horizontal skalieren, da jede Anforderung von einer beliebigen Instance im Pool oder Cluster verarbeitet werden kann. Mittels Auto Scaling können Sie einfach weitere Ressourcen nach Bedarf hinzufügen. Zustandsprüfungskriterien ermöglichen darüber hinaus den programmatischen Umgang mit wechselnden Datenverarbeitungs-, Kapazitäts- und Durchsatzanforderungen. Eine zustandslos konzipierte Anwendung könnte auch für eine serverlose Architektur umgeschrieben werden, mit Lambda-Funktionen anstelle von EC2 Instances. Lambda-Funktionen verfügen übrigens auch über eine integrierte dynamische Skalierung.

Für den Fall, dass eine Anwendungsressource wie ein Web-Server unbedingt Zustandsinformationen über Transaktionen benötigt, können Sie Ihre Anwendungen so entwickeln, dass die zustandsbehafteten Teile der Anwendung von den Servern entkoppelt sind. Ein HTTP-Cookie oder ähnliche Zustandsdaten ließen sich zum Beispiel in einer Datenbank speichern, vielleicht in DynamoDB oder in einem S3-Bucket oder EBS-Volume.

Wenn Sie einen komplexen mehrphasigen Workflow haben, ist es nicht erforderlich, den aktuellen Status jeder einzelnen Workflow-Phase nachzuverfolgen. Amazon Simple Workflow Service (SWF) kann den Ausführungsverlauf zentral speichern und diese Workloads zustandslos machen.

Eine weitere Maßnahme zum Erreichen von Ausfallsicherheit ist die verteilte Verarbeitung. Für Anwendungsfälle, bei denen große Datenmengen verarbeitet werden müssen, und das in einem angemessenen Zeitraum, in dem eine Datenverarbeitungsressource allein die Aufgabe nicht bewältigen kann, können Sie Ihre Workloads so anlegen, dass Aufgaben und Daten in kleinere Fragmente unterteilt werden, um diese in einem Cluster mit Datenverarbeitungsressourcen parallel zu verarbeiten. Die verteilte Verarbeitung ist zustandslos, da die unabhängigen Knoten, auf denen die unterteilten Daten und Aufgaben verarbeitet werden, ausfallen könnten. In diesem Fall würden die fehlgeschlagenen Aufgaben automatisch auf einem anderen Knoten im

verteilten Verarbeitungs-Cluster gestartet und wieder in die Scheduling Engine für die verteilte Verarbeitung eingegliedert.

AWS bietet eine Vielzahl von verteilten Datenverarbeitungs-Engines wie Amazon EMR, Amazon Athena und Amazon Machine Learning. Alle sind Managed Services, die Endpunkte bereitstellen und Sie von komplexen Aufgaben wie Patching, Wartung, Skalierung, Failover usw. entlasten.

Für die Echtzeitverarbeitung von Streaming-Daten kann Amazon Kinesis Streams die Daten in mehrere Shards unterteilen, die dann von mehreren Datenverbrauchern verarbeitet werden können, z. B. von Lambda-Funktionen oder EC2 Instances.

Weitere Informationen zu diesen Arten von Workloads finden Sie unter Big Data-Analyseoptionen auf AWS.[13]

Multi-Zone und Multi-Region

AWS-Produkte werden an zahlreichen Standorten auf der ganzen Welt gehostet. Diese Standorte bestehen aus Regionen und Availability Zones. Eine Region ist eine separater geografischer Bereich. Jede Region verfügt über mehrere isolierte Standorte, die so genannten Availability Zones. AWS bietet Kunden die Möglichkeit, Ressourcen, z. B. Instances, und Daten an mehreren Standorten zu platzieren.

Konzeptionieren Sie Ihre Anwendungen so, dass sie über mehrere Availability Zones und Regionen verteilt sind. Zusätzlich zur Verteilung und Replikation von Ressourcen über mehrere Availability Zones und Regionen sollten Sie Ihre Anwendungen so entwickeln, dass sie Load Balancer und Failover-Mechanismen verwenden. Auf diese Weise können Ihre Anwendungs-Stacks Datenflüsse und den Datenverkehr bei einem Ausfall automatisch auf diese alternativen Standorten umleiten.

Load Balancing

Mit dem Elastic Load Balancing Service (ELB) kann eine Gruppe von Anwendungs-Servern einem Load Balancer zugeordnet werden und dennoch auf mehrere Availability Zones verteilt sein. Wenn die EC2 Instances in einer bestimmten Availability Zone hinter einem Load Balancer ihre Zustandsprüfungen nicht bestehen, leitet der Load Balancer den Datenverkehr nicht mehr an diese Knoten. Sofern vorhanden, sorgt Auto Scaling dafür, dass

die Anzahl fehlerfreier Knoten automatisch mit den anderen Availability Zones ausgeglichen wird, ganz ohne manuelles Zutun.

Mit Amazon Route 53 und latenzbasierten DNS-Routing-Algorithmen lässt sich Load Balancing auch regionenübergreifend nutzen. Weitere Informationen finden Sie unter Latenzbasiertes Routing.[14]

Load Shredding-Strategien

Unter dem so genannten *Load Shredding* in cloudbasierten Infrastrukturen versteht man die Umleitung (u. a. per Proxyvorgang) von Datenverkehr an einen anderen Ort, um die Primärsysteme zu entlasten. In manchen Fällen besteht die Load Shredding-Strategie aus nicht mehr als einer bloßen Selektierung. Dabei entscheiden Sie, bestimmte Datenströme oder Datenverkehr nicht weiter zu übertragen, oder die Funktionalität Ihrer Anwendungen zu reduzieren, um die Rechenlast zu verringern, damit zumindest ein Teil der eingehenden Anfragen verarbeitet werden kann.

Es gibt zahlreiche Techniken, die für das Load Shredding verwendet werden können. Eine Methode ist das latenzbasierte DNS-Routing. Eine weitere Methode ist Caching. Caching kann in der Nähe der Anwendung über eine In-Memory-Caching-Ebene erfolgen, wie z. B. Amazon ElastiCache. Oder Sie verwenden eine Caching-Ebene, die sich näher am Edge des Benutzers befindet. Das funktioniert über ein globales CDN (Content Delivery Network, Netzwerk zur Bereitstellung von Inhalten) wie Amazon CloudFront.

Weitere Informationen über ElastiCache und CloudFront finden Sie unter „Erste Schritte mit ElastiCache“[15] und Amazon CloudFront CDN.[16]

Kostenoptimierung

Reserved-, Spot- oder On-Demand-Instances

Eng an die Fähigkeit zur dynamischen Bereitstellung von Ressourcen in der Cloud basierend auf Systemmetriken und anderen Performance- und Zustandsprüfungskriterien gebunden ist die Möglichkeit, die Kosten der Bereitstellung von Ressourcen in der Cloud zu kontrollieren. Mit Auto Scaling kann die Ressourcennutzung sehr genau auf die tatsächlichen Verarbeitungs- und Speicheranforderungen abgestimmt werden, sodass unnötige Kosten und unterausgelastete Ressourcen vermieden werden.

Eine andere Möglichkeit zur Kostenkontrolle in der Cloud bietet die Auswahl von On-Demand-Instances, Reserved Instances (RIs) oder Spot-Instances. Es gibt auch die Möglichkeit, Kapazität für DynamoDB zu reservieren.

Bei On-Demand-Instances zahlen Sie nur für die EC2 Instances, die Sie tatsächlich nutzen. Sie zahlen ohne langfristige Bindung stundenbasiert für die bereitgestellte Rechenkapazität.

Amazon EC2 Reserved Instances sind im Vergleich zu On-Demand-Instances bis zu 75 % kostengünstiger und bieten den zusätzlichen Vorteil der Reservierung von Kapazität in einer bestimmten Availability Zone. Neben dem Rabatt und der Reservierung von Verfügbarkeit gibt es keine funktionellen Unterschiede zwischen Reserved Instances und On-Demand-Instances.

Mit Spot-Instances können Sie auf nicht genutzte Amazon-EC2-Rechenkapazität bieten. Spot-Instances sind häufig zu einem niedrigeren Preis verfügbar sind (verglichen mit On-Demand-Instances). Dadurch können Sie Ihre cloudbasierten Anwendungen kostengünstiger betreiben.

Wenn Sie für die Cloud planen, eignen sich einige Anwendungsfälle besser für die Verwendung von Spot-Instances als andere. Beispiel: Da Spot-Instances jederzeit gelöscht werden können, sobald der Gebotspreis Ihr Gebot überschreitet, sollten Sie Spot-Instances nur für relativ zustandslose und horizontal skalierte Anwendungs-Stacks verwenden. Für zustandsbehaftete Anwendungen oder teure Verarbeitungslasten sind Reserved Instances und On-Demand-Instances wahrscheinlich besser geeignet. Für geschäftskritische Anwendungen, bei denen Kapazitätsbeschränkungen keine Rolle spielen, sind Reserved Instances die beste Wahl.

Weitere Informationen finden Sie unter Reserved Instances[17] und Spot-Instances[18].

Ereignismanagementprozess

Die Vorbereitung auf ein Infrastrukturereignis erfordert die Beteiligung von Anwendungsentwicklern, Administratoren und sonstigen Beteiligten aus dem Unternehmen. Sie sollten bereits Wochen vor einem Infrastrukturereignis regelmäßige Besprechungen mit den zentralen technischen Mitarbeitern

planen, die für den Betrieb der wichtigsten Infrastrukturkomponenten des Web-Service zuständig sind.

Der Zeitplan eines Infrastrukturereignisses

Die Planung eines Infrastrukturereignisses sollte mehrere Wochen vor dem Datum des Ereignisses beginnen. Eine typische Zeitleiste im Lebenszyklus eines geplanten Ereignisses ist in Abbildung 2 dargestellt.

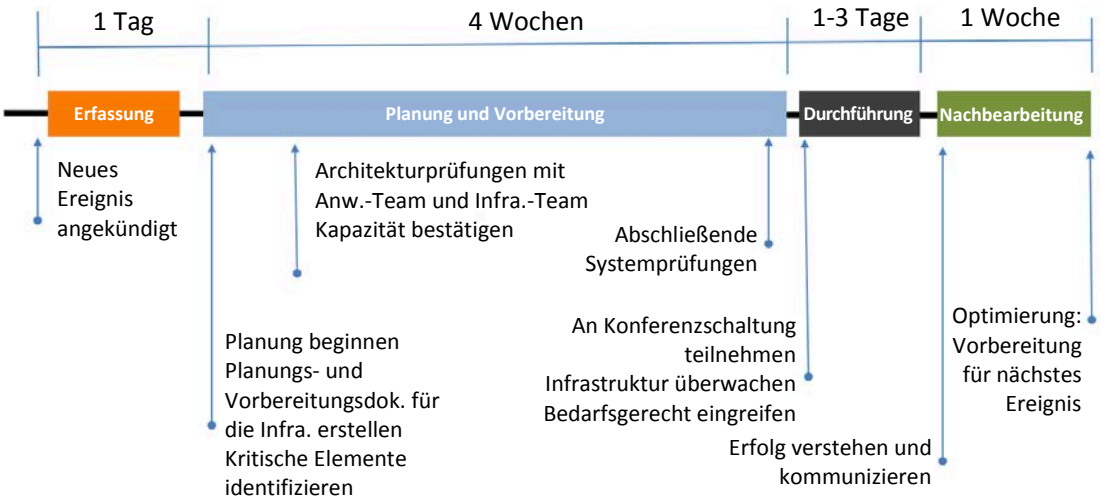

Abbildung 2. Zeitleiste eines typischen Infrastrukturereignisses

Planung und Vorbereitung

Zeitplan

Wir empfehlen in den Wochen vor einem Infrastrukturereignis den folgenden Aktivitätenzeitplan:

Woche 1:

- Stellen Sie ein Team zusammen, das für die Planung und Entwicklung des Infrastrukturereignisses verantwortlich ist.
- Führen Sie Besprechungen mit den beteiligten Personen durch, um einen Überblick über die Parameter des Ereignisses (Skalierung, Dauer, Zeitpunkt, geografische Reichweite, betroffene Workloads) und die Erfolgskriterien zu gewinnen.

- Involvieren Sie alle Downstream- oder Upstream-Partner und -Anbieter.

Woche 2-3:

- Überprüfen Sie die Architektur und nehmen Sie die nötigen Anpassungen vor.
- Führen Sie eine betriebliche Überprüfung durch und nehmen Sie die nötigen Anpassungen vor.
- Befolgen Sie die bewährten Methoden, die in diesem Dokument und in den Fußnoten beschrieben werden.
- Identifizieren Sie Risiken und entwickeln Sie Pläne zu deren Minderung.
- Entwickeln Sie ein Runbook für geplante Ereignisse.

Woche 4:

- Überprüfen Sie alle Cloud-Anbieter-Services, die angesichts der voraussichtlichen Last skaliert werden müssen.
- Prüfen Sie die Service Limits und erhöhen Sie diese bei Bedarf.
- Richten Sie ein Überwachungs-Dashboard und Warnungen für festgelegte Schwellenwerte ein.

Prüfung der Architektur

Ein wesentlicher Bestandteil der Vorbereitung auf ein Infrastrukturereignis ist eine strukturelle Prüfung der Anwendungs-Stacks, die den Zuwachs des Datenverkehrs verzeichnen werden. Der Zweck der Prüfung ist die Überprüfung und Ermittlung potenzieller Risikobereiche in Bezug auf die Skalierbarkeit oder die Zuverlässigkeit der Anwendung. Außerdem sollen dabei Optimierungsmöglichkeiten im Vorfeld des Ereignisses identifiziert werden.

Enterprise Support-Kunden bietet AWS ein Framework für die Überprüfung von Kundenanwendungs-Stacks, das sich auf fünf Designsäulen stützt. Diese Säulen sind Sicherheit, Zuverlässigkeit, Leistungseffizienz, Kostenoptimierung und betriebliche Spitzenleistung, wie unten beschrieben.

Tabelle 1: Säulen optimal strukturierter Anwendungen

Name der Säule	Definition der Säule	Relevanter Interessenbereich
Sicherheit	Die Fähigkeit zum Schutz von Informationen, Systemen und Komponenten bei gleichzeitiger Bereitstellung eines geschäftlichen Mehrwerts durch Risikobewertungen und Risikominderungsstrategien.	Identitätsmanagement, Verschlüsselung, Überwachung, Protokollierung, Schlüsselmanagement, Dedicated Instances, Compliance und Governance
Zuverlässigkeit	Die Fähigkeit eines Systems, den Betrieb nach Infrastruktur- oder Serviceausfällen wiederherzustellen, Datenverarbeitungsressourcen nach Bedarf dynamisch zu beziehen und die Auswirkung von Unterbrechungen wie Fehlkonfigurationen oder vorübergehenden Netzwerkproblemen zu mindern.	Service Limits, mehrere Availability Zones und Regionen, Skalierbarkeit, Zustandsprüfung/Überwachung, Sicherung/Notfallwiederherstellung, Netzwerke, automatische Fehlerkorrektur
Leistungseffizienz	Die Fähigkeit, Datenverarbeitungsressourcen effizient einzusetzen, um Systemanforderungen zu erfüllen, und diese Effizienz bei Änderungen von Bedarf und Technologien beizubehalten.	Die richtigen AWS-Produkte, Ressourcenauslastung, Speicherarchitektur, Caching, Latenzanforderungen
Kostenoptimierung	Die Fähigkeit, unnötige Kosten oder suboptimale Ressourcen zu umgehen oder zu beseitigen.	Spot-/Reserved Instances, Umgebungsoptimierung, Serviceauswahl, Volume-Optimierung, Kontoverwaltung, konsolidierte Fakturierung, Außerbetriebnahme von Ressourcen
Betriebliche Spitzenleistung	Die Fähigkeit zum Ausführen und Überwachen von Systemen, die einen geschäftlichen Mehrwert bieten, sowie zur kontinuierlichen Verbesserung geschäftsfördernder Prozesse und Verfahren.	Runbooks, Playbooks, Corporate Identity/Corporate Design, Game Days, Infrastruktur als Code, Fehler-Ursachen-Analysen

Im Anhang dieses Whitepapers finden Sie eine detaillierte Checkliste mit strukturellen Prüfpunkten. Diese kann verwendet werden, um einen AWS-basierten Anwendungs-Stack zu überprüfen.

Betriebliche Prüfung

Neben einer strukturellen Prüfung, deren Schwerpunkt eher auf den Designkomponenten einer Anwendung liegt, sollten Sie eine Prüfung Ihres Cloud-Betriebs und Ihrer Cloud-Managementverfahren durchführen, um die Verwaltungseffizienz Ihrer Cloud-Workloads zu beurteilen. Das Ziel dieser Prüfung besteht darin, betriebliche Defizite und Probleme zu identifizieren, um im Vorfeld des Ereignisses entsprechende Maßnahmen zu ergreifen.

Enterprise Support-Kunden bietet AWS eine Prüfung des Cloud-Betriebs. Diese kann als nützliches Hilfsmittel zur Vorbereitung auf ein Infrastrukturereignis eingesetzt werden. Die Prüfung konzentriert sich auf die Bewertung der folgenden Bereiche:

- Bereitschaft – Sie müssen über die richtige Kombination aus Organisationsstruktur, Prozessen und Technologien verfügen. Sie sollten klare Rollen und Verantwortlichkeiten für die Mitarbeiter etabliert haben, die Ihren Anwendungs-Stack verwalten. Prozesse sollten im Voraus auf das Ereignis abgestimmt werden. Verfahren sollten wenn möglich automatisiert werden.
- Überwachung – Effektive Überwachung misst die Performance einer Anwendung. Überwachung ist wichtig, um Anomalien zu erkennen, bevor sie zu Problemen werden, und bietet Möglichkeiten zur Minimierung der Auswirkungen unerwünschter Ereignisse.
- Betrieb – Betriebliche Aktivitäten müssen rechtzeitig und zuverlässig durchgeführt werden und sollten nach Möglichkeit automatisiert werden. Dies sollte auch den Umgang mit unerwarteten betrieblichen Ereignissen umfassen, die eine Eskalation erfordern.
- Optimierung – Führen Sie eine Post-mortem-Analyse mit erfassten Metriken, betrieblichen Trends und Erfahrungswerten durch, um bei künftigen Ereignissen Verbesserungsmöglichkeiten erfassen und melden zu können. Aus der Kombination von Optimierung und Vorbereitung entsteht eine Feedback-Schleife, die Sie unterstützt, Probleme mit dem Betrieb zu bewältigen und zu verhindern, dass diese erneut auftreten.

AWS Service Limits

Während eines geplanten Infrastrukturereignisses sollten Sie unbedingt vermeiden, bei der Skalierung einer Anwendung oder Workload die Service Limits zu überschreiten, die durch einen Cloud-Anbieter vorgegeben wurden.

Anbieter von Cloud-Services legen in der Regel Beschränkungen für die verschiedenen Ressourcen fest, die Sie verwenden können. Diese Beschränkungen gelten üblicherweise pro Konto und pro Region. Davon sind unter anderem folgende Ressourcen betroffen: Instances, Volumes, Streams, serverlose Aufrufe, Snapshots, die Anzahl der VPCs und Sicherheitsregeln. Die Beschränkungen dienen als Sicherheitsmaßnahme gegen eine unkontrollierte

Codeausführung oder den versuchten Ressourcenmissbrauch durch kriminelle Entitäten sowie zur Minimierung des Fakturierungsrisikos.

Manche Service Limits werden mit der Zeit automatisch erhöht, wenn Sie Ihre Präsenz in der Cloud erweitern. Die meisten dieser Services erfordern jedoch, dass Sie einen Support-Vorgang eröffnen, um eine Anfrage zur Erhöhung der Beschränkung zu stellen. Viele Service Limits können auf diese Weise erhöht werden, aber es gibt auch Services, deren Beschränkungen nicht geändert werden können.

AWS bietet Enterprise und Business Support-Kunden Trusted Advisor. Dieses stellt den Kunden ein Limit Check-Dashboard für die proaktive Verwaltung aller Service Limits bereit.

Weitere Informationen zu den Beschränkungen und der Verwaltung verschiedener AWS-Produkte finden Sie unter Trusted Advisor[19] und AWS Service Limits.[20]

Verstehen von Mustern

Baselines

Vor dem Beginn eines Infrastrukturereignisses sollten Sie „Normal“-Werte für wichtige Metriken dokumentieren. Auf diese Weise können Sie feststellen, wann eine Anwendung oder ein Service nach dem Abschluss/Ende des Ereignisses wieder in den Normalzustand zurückgekehrt ist. Wenn Sie beispielsweise ermitteln, dass die übliche Transaktionsrate über einen Load Balancer 2 500 Anfragen pro Sekunde ist, können Sie den Zeitpunkt nach dem Ereignis bestimmen, an dem Sie die Abwicklungsverfahren problemlos einleiten können.

Datenflüsse und Abhängigkeiten

Wenn Sie verstehen, wie Daten durch die verschiedenen Komponenten einer Anwendung fließen, können Sie potenzielle Engpässe und Abhängigkeiten besser ermitteln. Weisen die datenverarbeitenden Anwendungs-Tiers oder -komponenten in einem Datenfluss die richtige Größe auf und ist eine passende automatische Skalierung eingerichtet, falls die datenproduzierenden Tiers oder Komponenten in einem Anwendungs-Stack hochskalieren? Können Daten bei einem Komponentenausfall in eine Warteschlange gestellt werden, bis die Komponente wiederhergestellt ist? Können Anbieter oder Verbraucher von Downstream- oder Upstream-Daten in Reaktion auf das Ereignis skaliert werden?

Verhältnismäßigkeit

Ein weiterer Aspekt, den Sie bei der Vorbereitung auf ein Infrastrukturereignis prüfen sollten, ist die Verhältnismäßigkeit der Skalierung, die die verschiedenen Komponenten eines Anwendungs-Stacks erfordern. Diese Verhältnismäßigkeit beträgt nicht immer 1:1. Eine 10-fache Steigerung der Transaktionen pro Sekunde in einem Load Balancer könnte z. B. eine 20-fache Steigerung der Speicherkapazität oder der Anzahl von Streaming-Shards oder Lese- und Schreibvorgänge in einer Datenbank erfordern – dies hängt von der Verarbeitung ab, die ggf. auf Verbraucherseite der Anwendung stattfindet.

Kommunikationsplan

Im Vorfeld des Ereignisses sollten Sie einen Kommunikationsplan aufstellen. Erstellen Sie eine Liste der internen Beteiligten und Support-Gruppen und legen Sie fest, wer in den verschiedenen Phasen des Ereignisses und in unterschiedlichen Szenarien kontaktiert werden soll, z. B. am Anfang, während und am Ende des Ereignisses, bei der Analyse im Anschluss an das Ereignis, Notfallkontakte, Ansprechpartner für die Fehlerbehebung usw.

Als Ansprechpartner können folgende Personen und Gruppen ausgewählt werden:

- Projektbeteiligte
- Betriebsleiter
- Entwickler
- Supportteams
- Teams von Cloud-Service-Anbietern
- Network Operations Center(NOC)-Team

Während der Erstellung einer Liste mit internen Ansprechpartnern sollten Sie außerdem eine Kontaktliste mit externen Beteiligten zusammentragen, die in die kontinuierliche Live-Bereitstellung der Anwendung involviert sind. Dies beinhaltet Partner und Anbieter, die für wichtige Komponenten des Stacks verantwortlich sind sowie Downstream- und Upstream-Anbieter, die externe Services, Daten-Feeds, Authentifizierungs-Services und mehr bereitstellen.

Diese Liste mit externen Beteiligten sollte außerdem folgende Kontakte umfassen:

- Infrastruktur-Hosting-Anbieter
- Telekommunikationsanbieter
- Live-Daten-Streaming-Partner
- PR-Marketing-Kontakte
- Werbepartner
- Technische Berater, die an der Service-Gestaltung beteiligt sind

Fordern Sie von jedem Anbieter folgende Informationen an:

- Live-Ansprechpartner für die Dauer des Ereignisses
- Kontaktdaten für Notfall-Support und Eskalationsprozesse
- Name, Telefonnummer und E-Mail-Adresse
- Bestätigung, dass technische Live-Kontakte verfügbar sein werden

AWS-Kunden mit Enterprise Support sind außerdem Technical Account Manager (TAMs, Technische Kundenberater) zugewiesen, die koordinieren und sicherstellen, dass dedizierte AWS-Support-Mitarbeiter bereitstehen, die mit dem Ereignis vertraut sind und Support bieten können. TAMs stehen auch während des Ereignisses auf Abruf bereit, nehmen an Besprechungen teil und unterstützen bei Bedarf die Support-Eskalation.

NOC-Vorbereitung

Im Vorfeld des Ereignisses sollten Sie Ihr Betriebs- bzw. Entwicklerteam anweisen, ein Dashboard für Live-Metriken zu erstellen, das während des Ereignisses für die Überwachung jeder wichtigen Komponente des im Produktionsbetrieb befindlichen Web-Service sorgt. Idealerweise sollte das Dashboard im Rahmen des Ereignisses automatisch jede Minute, oder in einem anderen geeigneten Intervall, aktualisierte Metriken anzeigen.

Wir empfehlen eine Überwachung der folgenden Elemente:

- Ressourcennutzung jedes Servers (CPU-, Festplatten- und Speicherauslastung)
- Reaktionszeit des Web-Service
- Metriken zum Web-Datenverkehr (Benutzer, Seitenaufrufe, Sitzungen)

- Web-Datenverkehr pro Besucherregion (globale Kundensegmente)
- Auslastung des Datenbank-Servers
- Marketing-Flow-Conversion-Funnels wie Konvertierungsraten und Fallout-Prozentsatz
- Anwendungsfehlerprotokolle
- Frühwarnsystem

Mit Amazon CloudWatch können Sie die meisten dieser Metriken von AWS-Ressourcen in Form von benutzerdefinierten CloudWatch-Dashboards in einer zentralen Übersicht zusammenführen. Darüber hinaus können Sie benutzerdefinierte Metriken in CloudWatch importieren, wenn diese nicht bereits von AWS automatisch bereitgestellt werden. Weitere Informationen zu AWS-Überwachungstools und -funktionen finden Sie im Abschnitt „Überwachung“ dieses Whitepapers.

Vorbereitung des Runbooks

In Vorbereitung auf das Infrastrukturereignis sollten Sie ein Runbook erstellen. Ein *Runbook* ist ein betrieblicher Leitfaden, der eine Zusammenstellung der Verfahren und Vorgänge enthält, die Ihre Mitarbeiter im Rahmen des Ereignisses durchführen werden. Ereignis-Runbooks können aus vorhandenen Runbooks für routinemäßige Vorgänge und die Behandlung von Ausnahmen entstehen. In der Regel enthält ein Runbook Verfahren zum Starten, Anhalten, Überwachen und Debuggen eines Systems. Außerdem sollte es Verfahren für die Handhabung von unerwarteten Ereignissen und Notfallsituationen beschreiben.

Ein Runbook sollte folgende Abschnitte beinhalten:

- **Ereignisdetails**: Eine kurze Beschreibung des Ereignisses, der Erfolgskriterien, der Medienaufmerksamkeit, der Ereignisdaten und der Kontaktdaten der wichtigsten Beteiligten auf Kundenseite und von AWS.
- **Liste der AWS-Produkte**: Eine Aufzählung sämtlicher im Rahmen der Veranstaltung zu verwendenden AWS-Produkte. Enthält außerdem die erwartete Last für diese Services sowie betroffene Regionen und Konto-IDs.
- **Architektur- und Anwendungsprüfung**: Dokumentiert die Lasttestergebnisse, sämtliche Belastungspunkte im Infrastruktur- und

Anwendungsdesign, Ausfallsicherheitsmaßnahmen für die Workload, Single Points of Failure und mögliche Engpässe.

- **Betriebliche Prüfung**: Beschreibt die Überwachungskonfiguration, Integritätskriterien, Benachrichtigungsmechanismen und Service-Wiederherstellungsverfahren.
- **Vorbereitungs-Checkliste**: Beinhaltet Punkte wie Service Limits-Prüfungen, den Vorlauf von Anwendungs-Stack-Komponenten wie Load Balancers, die Bereitstellung vorkonfigurierter Ressourcen wie Streaming-Shards, DynamoDB-Partitionen, S3-Partitionen usw. Weitere Informationen finden Sie in der detaillierten Checkliste zur Überprüfung der Architektur im Anhang dieses Whitepapers.

Überwachung

Überwachungsplan

Die Überwachung von Datenbanken, Anwendungen und Betriebssystemen ist von entscheidender Bedeutung, um ein erfolgreiches Ereignis zu gewährleisten. Sie sollten umfassende Überwachungssysteme einrichten, damit Sie schwerwiegende Vorfälle, die während des Ereignisses auftreten, auf effektive Weise erfassen und sofortige Maßnahmen ergreifen können. Grundsätzlich sorgt eine effektive Überwachungsstrategie dafür, dass die Überwachungs-Tools für eine Anwendung je nach geschäftlicher Relevanz auf der geeigneten Ebene eingesetzt werden. Eine effektive Incident-Management-Strategie vereint AWS- und kundenseitige Überwachungsdaten mit den entsprechenden Tools und Prozessen für das Ereignis- und Incident-Management. Die Implementierung eines Überwachungsplans, der die Gesamtheit der Überwachungsdaten aus allen AWS-Lösungsbereichen zusammenführt, kann das Debugging im Falle eines komplexen Ausfalls wesentlich erleichtern.

Der Überwachungsplan sollte Antworten auf folgende Fragen enthalten:

- Welche Überwachungstools und -Dashboards müssen für das Ereignis eingerichtet werden?
- Was sind die Überwachungsziele und die zulässigen Schwellenwerte? Welche Ereignisse lösen Aktionen aus?
- Welche Ressourcen und welche Metriken dieser Ressourcen werden überwacht und wie oft müssen sie abgefragt werden?

- Wer führt die Überwachungsaufgaben durch? Welche Überwachungswarnungen liegen vor? Wer wird benachrichtigt?
- Welche Wiederherstellungspläne sind für typische und erwartete Ausfälle in Kraft? Wie wird mit unerwarteten Ereignissen verfahren?
- Wie sieht der Eskalationsprozess bei Ausfällen aus?

Im Rahmen dieser Strategie können folgende AWS-Überwachungstools verwendet werden:

- **Amazon CloudWatch**: Eine sofort einsetzbare Lösung für Dashboard-Metriken, Überwachung, Benachrichtigungen und automatisierte Bereitstellung mit AWS.
- **Benutzerdefinierte Amazon CloudWatch-Metriken**: Zur Erfassung von Betriebssystem-, Anwendungs- und Business-Metriken. Die Amazon CloudWatch-API ermöglicht die Erfassung von nahezu jeder Art benutzerdefinierter Metriken.
- **Amazon EC2 Instance-Zustand**: Zur Anzeige von Statusprüfungen und zum Planen von Ereignissen für Ihre Instances, basierend auf deren Status, z. B. automatischer Neustart oder Neustart einer Instance.
- **Amazon SNS**: Zum Einrichten, Ausführen und Senden von ereignisgesteuerten Benachrichtigungen.
- **AWS X-Ray**: Unterstützt das Debugging und die Analyse von verteilten Anwendungen und einer Microservices-Architektur durch die Datenflussanalyse zwischen Systemkomponenten.
- **Amazon Elasticsearch Service**: Zur zentralen Erfassung von Protokollen und Protokollanalyse in Echtzeit. Für eine schnelle, heuristische Erkennung von Problemen.
- **Drittanbieter-Tools:** Für Echtzeitanalysen, Full-Stack-Überwachung und -Sichtbarkeit.
- **Standard-Überwachungstools für Betriebssysteme**: Zur Überwachung auf Betriebssystemebene.

Weitere Informationen zu den AWS-Überwachungstools finden Sie unter Automatische und manuelle Überwachung.[21] Siehe auch Verwenden von Amazon CloudWatch-Dashboards[22] und Veröffentlichen benutzerdefinierter Metriken.[23]

Benachrichtigungen

Ein wesentliches betriebliches Element Ihres Designs für Infrastrukturereignisse ist die Konfiguration von Alarmen und Benachrichtigungen zur Integration in Ihre Überwachungslösungen. Diese Alarme und Benachrichtigungen können mit Services wie AWS Lambda genutzt werden, um Aktionen basierend auf der jeweiligen Warnung auszulösen. Die Automatisierung von Reaktionen auf betriebliche Ereignisse ermöglicht eine Minderung, ein Rollback und eine Wiederherstellung mit maximaler Reaktionsfähigkeit.

Darüber hinaus sollten Tools eingesetzt werden, um Workloads zentral zu überwachen und geeignete Warnungen und Benachrichtigungen zu erstellen, die auf den verfügbaren Protokollen und Metriken im Zusammenhang mit wichtigen betrieblichen Indikatoren basieren. Dazu zählen Warnungen und Benachrichtigungen für überschrittene Werte sowie Service- oder Komponentenausfälle. Idealerweise ist das System so konzipiert, dass es bei einer Überschreitung von Grenzwerten oder bei Ausfällen in Reaktion auf diese Benachrichtigungen und Warnungen eine automatische Fehlerkorrektur oder eine entsprechende Skalierung vornimmt.

Wie bereits erwähnt, bietet AWS Services (Amazon SQS und Amazon SNS), um sicherzustellen, dass entsprechende Warnungen und Benachrichtigungen in Reaktion auf ungeplante betriebliche Ereignisse ausgegeben werden, und um automatisierte Reaktionen zu ermöglichen.

Betriebliche Bereitschaft (Tag des Ereignisses)

Ausführung des Plans

Am Tag des Ereignisses sollte das Team, das die Hauptverantwortung für das Infrastrukturereignis trägt, über eine Telefonkonferenz miteinander in Verbindung stehen und die Live-Dashboards überwachen. Runbooks sollten vollständig ausgearbeitet und allgemein verfügbar sein. Stellen Sie sicher, dass der Kommunikationsplan eindeutig definiert und allen Support-Mitarbeitern und Beteiligten vertraut ist. Außerdem sollte ein Notfallplan vorhanden sein.

Konferenzschaltung

Sorgen Sie dafür, dass während des Ereignisses eine Live-Konferenzschaltung mit den folgenden Teilnehmern besteht:

- Die hauptverantwortlichen Anwendungs- und Betriebsteams
- Die Betriebsteamleitung
- Technische Ressourcen von externen Partnern, die direkt in die technische Bereitstellung involviert sind
- Beteiligte aus dem Unternehmen

Für weite Teile des Ereignisses sollte sich die Gesprächsbeteiligung der Teilnehmer dieser Konferenzschaltung auf ein Minimum beschränken. Sollte jedoch ein betriebliches Ereignis eintreten, das einen Eingriff erforderlich macht, dann sind die wichtigsten Entscheidungsträger bereits zugeschaltet und können sich mit Rat und Tat einbringen.

Berichterstattung für die Führungsebene

Senden Sie während des Ereignisses einmal pro Stunde eine E-Mail an die wichtigsten Beteiligten in Führungspositionen. Dieses Update sollte Folgendes umfassen:

- Statuszusammenfassung: Grün (auf Kurs), Gelb (es sind Probleme aufgetreten), Rot (großes Problem)
- Update der wichtigsten Metriken
- Aufgetretene Probleme, Status des Lösungsplans, geschätzte Zeit bis zur Lösung
- Telefonnummer der Konferenzschaltung (für etwaige weitere Teilnehmer)

Nach Abschluss des Ereignisses sollte eine zusammenfassende E-Mail in einem ähnlichen Format gesendet werden.

Notfallplan

Für jeden Schritt der Vorbereitung auf das Ereignis sollte ein entsprechender Rollback-Plan vorliegen, der in einer Testumgebung geprüft wurde.

Beachten Sie bei der Zusammenstellung eines Rollback-Plans die folgenden Fragen:

- Welche Worst-Case-Szenarien könnten während des Ereignisses eintreten?

- Welche Ereignisse würden negative PR-Auswirkungen haben?
- Welche Komponenten und Services von Drittanbietern könnten während des Ereignisses ausfallen?
- Welche Metriken würden auf den Eintritt eines unerwünschten Szenarios hinweisen und sollten deshalb überwacht werden?
- Wie sieht der Rollback-Plan für jedes mögliche Szenario aus?
- Wie lange dauern die Rollback-Prozesse? Was ist jeweils das akzeptable Recovery Point Objective (RPO) und Recovery Time Objective (RTO)? (Weitere Informationen zu diesen Konzepten finden Sie unter Verwenden von AWS für die Notfallwiederherstellung[24].)

Die folgenden Rollback-Arten sollten Ihnen vertraut sein:

- **Blue/Green-Bereitstellung**: Bei der Einführung einer neuen Produktions-App oder einer neuen Umgebung lassen Sie den vorherigen Produktions-Build online, sodass Sie bei Bedarf schnell wieder darauf zurückfallen können.
- **Warm Pilot**: Erstellen Sie eine minimale Umgebung in einer zweiten Region, die bei Bedarf schnell hochskaliert werden kann. Bei einem Ausfall der primären Region können Sie eine schnelle Skalierung in der Backup-Region vornehmen und den Datenverkehr auf die zweite Region umleiten.
- **Fehlerseiten im Wartungsmodus**: Prüfen Sie Einrichtung und Auslöser der Fehlerseiten auf jeder Ebene Ihres Web-Service. Sie sollten je nach Bedarf eine detailliertere Nachricht in diese Fehlerseiten einfügen können.

Testen und dokumentieren Sie die einzelnen Rollback-Pläne für jedes mögliche Ausfallszenario.

Nachbearbeitung

Post-mortem-Analyse

Eine Post-mortem-Analyse wird oftmals nicht in Betracht gezogen, da Kunden in der Regel darauf bedacht sind, den normalen Betrieb schnellstmöglich wieder aufzunehmen. Allerdings empfehlen wir, dass Sie eine Post-mortem-Analyse als Teil des Verwaltungslebenszyklus eines Infrastrukturereignisses durchführen.

Post-mortems ermöglichen Ihnen die Zusammenarbeit mit allen beteiligten Teams und die Identifizierung von Bereichen, die einer weiteren Optimierung bedürfen, wie etwa Betriebsabläufe, Implementierungsdetails, Failover- und Wiederherstellungsverfahren. Dies ist besonders wichtig, wenn während des Ereignisses bei einem Anwendungs-Stack Störungen aufgetreten sind. Eine Post-mortem-Analyse des Ereignisses ist auch für die Dokumentation hilfreich, wenn Dokumente zur Ursachenanalyse (RCA, Root Cause Analysis) erstellt werden müssen.

Abwicklungsprozess

Unmittelbar nach Beendigung der Infrastrukturereignisses sollte mit dem Abwicklungsprozess begonnen werden. Während dieses Zeitraums ist es ratsam, die Überwachung relevanter Anwendungen und Services fortzusetzen, um sicherzustellen, dass der Datenverkehr wieder auf das normale Produktionsniveau zurückgekehrt ist. Verwenden Sie alle Zustands-Dashboards, die während der Vorbereitungsphase erstellt wurden, um die Normalisierung des Datenverkehrs und der Transaktionsraten zu überprüfen. Die Abwicklungszeiträume einiger Ereignisse sind linear und unkompliziert. Andere hingegen weisen einen unregelmäßigen oder graduelleren Volume-Rückgang auf. Einige Datenverkehrsmuster bleiben unter Umständen bestehen. Die Wiederaufnahme des Betriebs nach einem Anstieg des Datenverkehrs erfordert beispielsweise zumeist unkomplizierte Abwicklungsprozesse. Eine Anwendungsbereitstellung oder Expansion in eine neue geografische Region kann jedoch langfristige Auswirkungen haben, weswegen Sie neue Datenverkehrsmuster sorgfältig überwachen müssen. Außerdem sollten zusätzliche Überwachungsmaßnahmen Bestandteil des permanenten Anwendungs-Stacks sein.

Irgendwann nach Beendigung des Ereignisses müssen Sie entscheiden, wann der Ereignis-Management-Betrieb gefahrlos beendet werden kann. Orientieren Sie sich an den zuvor dokumentierten „normalen“ Werten wichtiger Metriken, um zu bestimmen, wann ein Ereignis abgeschlossen oder beendet wurde. Wir empfehlen das Aufteilen von Abwicklungsaktivitäten in zwei Bereiche, die unterschiedliche Zeitpläne aufweisen können. Konzentrieren Sie den ersten Bereich auf die operative Verwaltung des Ereignisses, wie das Senden von Nachrichten an interne und externe Beteiligte und Partner und die Zurücksetzung von Service Limits. Konzentrieren Sie den zweiten Bereich auf die technischen Aspekte der Abwicklungsprozesse, wie Scale-down-Verfahren, die Validierung des Umgebungszustands und Kriterien zur Entscheidung

darüber, ob Änderungen an der Architektur zurückgesetzt oder übernommen werden sollen.

Der Zeitplan für den jeweiligen Bereich ist von der Art des Ereignisses, den wichtigsten Metriken und den Kundenvorstellungen abhängig. In der folgenden Tabelle haben wir einige allgemeine Aufgaben aufgelistet, die mit diesen Bereichen verbunden sind. So können Sie den passenden Zeitpunkt für das Ende des Managements eines Ereignisses bestimmen.

Tabelle 2: Operative Abwicklungsaufgaben

Aufgabe	Beschreibung
Kommunikation	Benachrichtigung interner und externer Beteiligter über die Beendigung des Ereignisses. Die Beendigungsmitteilung sollte mit dem definitiven Ende des Ereignisses zusammenfallen. Orientieren Sie sich an „normalen“ Metriken, um zu bestimmen, wann es angebracht ist, die Kommunikation zu beenden. Alternativ können Sie die Kommunikation stufenweise beenden. Sie können beispielsweise die ereignisspezifische Kommunikation auflösen, übliche Eskalationsverfahren jedoch noch beibehalten, falls es nach dem Ereignis zu einem Ausfall kommt.
Service Limits/ Kostenreduzierung	Obwohl es verlockend sein kann, nach einem Ereignis ein erhöhtes Service Limit aufrechtzuerhalten, sollten Sie bedenken, dass Service Limits auch als Absicherung dienen. Service Limits schützen Sie vor überhöhten Kosten, indem sie eine übermäßige Service-Nutzung vermeiden, ob diese durch ein kompromittiertes Konto oder eine falsch konfigurierte Automatisierung entsteht.
Berichterstellung und Analyse	Die Datensammlung und das Zusammentragen von Ereignismetriken sollten sorgfältig geplant und an alle im Kommunikationsplan genannten internen Parteien weitergeleitet werden. Auch Analysen mit Mustern, Trends, Problembereichen, erfolgreichen Verfahren, Ad-hoc-Verfahren, dem Zeitplan des Ereignisses und Informationen, ob Erfolgskriterien erfüllt wurden oder nicht, dürfen nicht fehlen. Des Weiteren sollte eine detaillierte Kostenanalyse entwickelt werden, um die betrieblichen Kosten für dieses Ereignis aufzuzeigen.
Optimierungsaufgaben	Unternehmensorganisationen entwickeln sich im Laufe der Zeit weiter, da sie ihr Aufgabenfeld kontinuierlich ausweiten. Operative Optimierung erfordert die konstante Aufzeichnung und Aufbewahrung von Metriken, betrieblichen Trends und Erkenntnissen, die bei Ereignissen gewonnen wurden. So ist es möglich, Verbesserungsgelegenheiten zu erkennen. Optimierung ist von der Vorbereitung einer Feedback-Schleife abhängig, mit der betriebliche Probleme identifiziert werden und für die Zukunft verhindert werden können.

Tabelle 3: Technische Abwicklungsaufgaben

Aufgabe	Beschreibung
Service Limits/ Kostenreduzierung	Obwohl es verlockend sein kann, nach einem Ereignis erhöhte Service Limits aufrechtzuerhalten, sollten Sie bedenken, dass Service Limits auch als Absicherung dienen. Service Limits schützen Ihren Betrieb und beschränken Ihre Betriebskosten, indem sie eine übermäßige Service-Nutzung vermeiden, ob diese durch bösartige Aktivitäten in einem kompromittierten Konto oder eine falsch konfigurierte Automatisierung hervorgerufen wird.
Scale-down-Verfahren	Das Zurücksetzen von Ressourcen, die während der Vorbereitungsphase hochskaliert wurden. Der Umfang ist von Ihrer Architektur abhängig, aber die folgenden Beispiele sind häufig: Instance-Größe EC2/RDS Auto Scaling-Konfiguration Reservierte Kapazität Bereitgestellte IOPS
Validierung des Umgebungszustands	Vergleich mit Ausgangsmetriken und Prüfung des Produktionszustandes. So wird sichergestellt, dass nach dem Ereignis und nach Abschluss des Scale-downs die betroffenen Systeme wie gewohnt funktionieren.
Entscheidung über Architekturänderungen	Einige Änderungen während der Vorbereitung des Ereignisses sollen möglicherweise beibehalten werden. Dies hängt von der Art des Ereignisses und den beobachteten Betriebsmetriken ab. Expansionen in neue Regionen können beispielsweise die Notwendigkeit einer permanenten Erhöhung von Ressourcen in der jeweiligen Region zur Folge haben. Auch kann die Erhöhung bestimmter Service Limits oder Konfigurationsparameter, wie der Anzahl der Partitionen in einer DB oder der Shards in einem PIOPS-Stream in einem Volume zur dauerhaften Leistungssteigerung genutzt werden.

Optimierung

Die vielleicht wichtigste Komponente der Infrastrukturereignisverwaltung ist die nachgelagerte Analyse und die Identifizierung von beobachteten Problemen beim Betrieb und der Architektur und Möglichkeiten zur Verbesserung. Infrastrukturereignisse sind nur selten einmalige Ereignisse. Sie können saisonabhängig sein oder mit der Veröffentlichung neuer Anwendungen zusammenfallen. Sie können allerdings auch durch das Wachstum eines Unternehmens auftreten, wenn dieses neue Märkte und Territorien erschließt. Daher ist jedes Infrastrukturereignis eine gute Gelegenheit, um Beobachtungen anzustellen, die eigene Methodik zu verfeinern und sich effektiver auf das nächste Ereignis vorzubereiten.

Fazit

AWS bietet elastische und programmierbare Bausteine in Form von Produkten und Services, die Ihr Unternehmen praktisch zur Unterstützung jeder Workload beliebig einsetzen kann. Mit den AWS-Ereignisrichtlinien und bewährten Methoden für Infrastrukturereignisse und unserer breiten Palette an hochverfügbaren Services kann Ihr Unternehmen Lösungen und Vorbereitungen für große Geschäftsereignisse ausarbeiten und sicherstellen, dass der Skalierungsbedarf problemlos und dynamisch erfüllt werden kann. So werden schnelle Reaktionen und eine globale Reichweite sichergestellt.

Mitwirkende

Dieses Dokument ist unter der Mitarbeit folgender Personen und Organisationen entstanden:

- Presley Acuna, AWS Enterprise Support Manager
- Kurt Gray, AWS Global Solutions Architect
- Michael Bozek, AWS Sr. Technical Account Manager
- Rovan Omar, AWS Technical Account Manager
- Will Badr, AWS Technical Account Manager
- Eric Blankenship, AWS Sr. Technical Account Manager
- Greg Bur, AWS Technical Account Manager
- Bill Hesse, AWS Sr. Technical Account Manager
- Hasan Khan, AWS Sr. Technical Account Manager
- Varun Bakshi, AWS Sr. Technical Account Manager

Weitere Informationen

Weitere Informationen zu bewährten Methoden bezüglich Betrieb und Architektur finden Sie unter Betriebschecklisten für AWS.[25] Wir empfehlen, dass Sie auch AWS Well Architected Framework[26] durchlesen, um einen strukturierten Ansatz zur Bewertung Ihrer cloudbasierten Anwendungsbereitstellungs-Stacks zu erhalten. AWS bietet Infrastrukturereignisverwaltung (IEM, Infrastructure Event Management) als

Premium Support-Angebot für Kunden, die eine direktere Einbeziehung der AWS Technical Account Manager und Support Engineers in ihre Konzeption, Planung und Betriebsabläufe bei Ereignissen wünschen. Weitere Informationen zum AWS IEM Premium Support-Angebot finden Sie unter Infrastructure Event Management.[27]

Anhang

Detaillierte Checkliste zur Architekturüberprüfung

Ja-Nein-N/V	Sicherheit
☐-☐-☐	Wir wechseln unsere AWS Identity and Access Management(IAM)-Zugangsschlüssel und -Benutzerpasswörter sowie die Anmeldeinformationen für die mit unserer Anwendung zusammenhängenden Ressourcen maximal alle drei Monate, gemäß den bewährten Methoden für AWS-Sicherheit. Für jedes Konto gilt die Passwortrichtlinie. Wir verwenden Hardware- oder virtuelle MFA-Geräte (Multifaktor-Authentifizierung).
☐-☐-☐	Wir verfügen über interne Sicherheitsprozesse und -kontrollen zur Steuerung von individuellen, rollenbasierten Zugriffen nach dem Konzept der geringsten Rechte auf AWS-APIs, die IAM nutzen.
☐-☐-☐	Wir haben alle vertraulichen Informationen, einschließlich eingebetteter öffentlicher/privater Instance-Schlüsselpaare entfernt. Außerdem haben wir alle SSH-genehmigten Schlüsseldateien aller angepassten Amazon Machine Images (AMIs) überprüft.
☐-☐-☐	Wir verwenden IAM-Rollen für EC2 Instances nach Bedarf, anstatt Anmeldeinformationen in AMIs einzubetten.
☐-☐-☐	Wir trennen IAM-Administratorberechtigungen von normalen Benutzerberechtigungen, indem wir eine IAM-Administratorrolle erstellen und die IAM-Aktionen anderer funktionaler Rollen einschränken.
☐-☐-☐	Wir wenden die neuesten Sicherheits-Patches auf unsere EC2 Instances für Windows- oder Linux-Instances an. Wir verwenden Betriebssystem-Zugriffskontrollen, darunter Amazon EC2-Sicherheitsgruppenregeln, VPC-Netzwerk-Zugriffskontrolllisten, Betriebssystemstabilisierung, hostbasierte Firewall, Systeme zum Erkennen/Verhindern von Eindringversuchen, Überwachung der Softwarekonfiguration und des Host-Bestands.
☐-☐-☐	Wir stellen sicher, dass die Netzwerkkonnektivität zu und von den AWS des Unternehmens und der AWS-Umgebung einen Transport von Verschlüsselungsprotokollen verwendet.
☐-☐-☐	Wir wenden eine zentrale Protokoll- und Audit-Management-Lösung zur Identifizierung und Analyse ungewöhnlicher Zugriffsmuster oder bösartiger Angriffe auf die Umgebung an.

Ja-Nein-N/V	Sicherheit
☐-☐-☐	Wir verfügen über Prozesse für Sicherheitsereignisse, Störfallmanagement, zur Korrelation und zur Berichterstellung.
☐-☐-☐	Wir stellen sicher, dass es keinen uneingeschränkten Zugriff auf AWS-Ressourcen in unseren Sicherheitsgruppen gibt.
☐-☐-☐	Wir wenden sichere Protokolle (HTTPS oder SSL), aktuelle Sicherheitsrichtlinien und Chiffren-Protokolle für eine Front-End-Verbindung (Client Load Balancer) an. Die Anfragen werden zwischen Client und Load Balancer verschlüsselt. Dieser Ansatz bietet größere Sicherheit.
☐-☐-☐	Wir konfigurieren unseren Amazon Route 53 MX-Ressourcendatensatz auf einen TXT-Ressourcendatensatz, der einen entsprechenden Sender Policy Framework(SPF)-Wert zur Spezifikation der Server aufweist, die dazu berechtigt sind, für unsere Domäne E-Mails zu versenden.

Ja-Nein-N/V	Zuverlässigkeit
☐-☐-☐	Wir stellen unsere Anwendung über eine Flotte von EC2 Instances bereit, die in einer Auto Scaling-Gruppe bereitgestellt werden, um eine automatische horizontale Skalierung basierend auf vordefinierten Skalierungsplänen sicherzustellen. Weitere Informationen.
☐-☐-☐	Wir nutzen eine Elastic Load Balancing-Zustandsprüfung in der Konfiguration unserer Auto Scaling-Gruppe, um sicherzustellen, dass die Auto Scaling-Gruppe gemäß dem Zustand der zugrunde liegenden EC2 Instances agiert. (Gilt nur, wenn Sie Load Balancer in Auto Scaling-Gruppen verwenden.)
☐-☐-☐	Wir haben kritische Komponenten unserer Anwendungen über mehrere Availability Zones verteilt bereitgestellt und führen angemessene Replikationen der Daten zwischen den Zonen durch. Wir testen, wie ein Ausfall innerhalb dieser Komponenten sich auf die Anwendungsverfügbarkeit mit Elastic Load Balancing, Amazon Route 53 oder geeigneten Drittanbieter-Tools auswirkt.
☐-☐-☐	Wir stellen unsere Amazon RDS-Instances auf Datenbankebene in mehreren Availability Zones bereit, um die Datenbankverfügbarkeit durch die synchrone Replikation von Daten auf eine Standby-Instance in einer anderen Availability Zone zu verbessern.
☐-☐-☐	Wir haben Prozesse für automatische oder manuelle Failover-Prozesse für Ausfälle oder Leistungseinbrüche definiert.
☐-☐-☐	Wir verwenden CNAME-Records für die Zuordnung zwischen unserem DNS-Namen und unseren Services. Wir verwenden KEINE A-Datensätze.
☐-☐-☐	Wir haben einen niedrigeren TTL-Wert (Time-to-live) für unseren Amazon Route 53-Datensatz konfiguriert. So werden Verzögerungen bei der Umleitung von Datenverkehr vermieden, wenn DNS-Auflöser aktualisierte DNS-Datensätze anzufordern. (Dies kann beispielsweise auftreten, wenn DNS Failover den Ausfall eines Ihrer Endpunkte erkennt und darauf reagiert.)
☐-☐-☐	Wir verfügen über mindestens zwei VPN-Tunnel, die darauf konfiguriert sind, bei

Ja-Nein-N/V	Zuverlässigkeit
	Ausfällen oder geplanten Wartungen der Geräte am AWS-Endpunkt Redundanz bereitzustellen.
☐-☐-☐	Wir verwenden AWS Direct Connect und verfügen über zwei Direct Connect-Verbindungen, die darauf konfiguriert sind, jederzeit Redundanz zur Verfügung zu stellen, wenn ein Gerät nicht verfügbar ist. Die Verbindungen werden für verschiedene Direct Connect-Standorte bereitgestellt, um Redundanz zur Verfügung zu stellen, falls ein Speicherort nicht verfügbar ist. Wir haben außerdem die Konnektivität zu unserem Virtual Private Gateway auf mehrere virtuelle Schnittstellen über mehrere Direct Connect-Verbindungen und Standorte konfiguriert.
☐-☐-☐	Wir verwenden Windows-Instances. Außerdem stellen wir sicher, dass wir die neuesten PV-Treiber verwenden. PV-Treiber unterstützen die Optimierung der Treiberleistung und die Minimierung von Laufzeitproblemen und Sicherheitsrisiken. Des Weiteren haben wir sichergestellt, dass der EC2Config-Agent die neueste Version unserer Windows-Instance ausführt.
☐-☐-☐	Wir erstellen Snapshots unserer Amazon Elastic Block Store(EBS)Volumes, um bei einem Ausfall eine zeitpunktbezogene Wiederherstellung zu gewährleisten.
☐-☐-☐	Wir verwenden, falls angemessen, separate Amazon EBS-Volumes für Betriebssystem- und Anwendungs-/Datenbankdaten.
☐-☐-☐	Wir verwenden für alle Linux-Instances die neuesten Kernel, Software- und Treiber-Patches.

Ja-Nein-N/V	Leistungseffizienz
☐-☐-☐	Vor ihrer Inbetriebnahme prüfen wir unsere in AWS gehosteten Anwendungskomponenten umfassend, auch auf ihre Leistung hin. Wir führen zudem Lasttests durch, um sicherzustellen, dass wir die richtige EC2 Instance-Größe, Anzahl von IOPS, RDS DB-Instance-Größe usw. verwendet haben.
☐-☐-☐	Wir führen einen Nutzungsprüfungsbericht über unsere Service Limits und stellen sicher, dass die aktuelle Nutzung aller AWS-Produkte höchstens 80 % des Service Limits beträgt. Weitere Informationen
☐-☐-☐	Wir verwenden ein CDN (Content Delivery/Distribution Network, Netzwerk zur Bereitstellung und Verteilung von Inhalten) für die Nutzung von Caching für unsere Anwendung (Amazon CloudFront) und zur Optimierung der Bereitstellung der Inhalte und ihrer automatischen Verteilung an den Edge-Standort, der dem Benutzer am nächsten ist.
☐-☐-☐	Wir wissen, dass einige dynamische HTTP-Anforderungs-Header, die von Amazon CloudFront empfangen werden (User-Agent, Datum usw.), die Leistung beeinträchtigen können, da sie die Cache-Trefferrate reduzieren und die Ursprungsbelastung erhöhen. Weitere Informationen
☐-☐-☐	Wir stellen sicher, dass der maximale Durchsatz einer EC2 Instance größer als der

Ja-Nein-N/V	Leistungseffizienz
	maximale aggregierte Durchsatz des verbundenen EBS-Volumes ist. Darüber hinaus nutzen wir EBS-optimierte Instances mit PIOPS-EBS-Volumes, um die erwartete Leistung der Volumes zu erhalten.
☐-☐-☐	Wir stellen sicher, dass der Lösungsentwurf keinen Engpass in der Infrastruktur oder einen Spannungspunkt in der Datenbank oder dem Anwendungskonzept aufweist.
☐-☐-☐	Wir stellen Überwachungs- und Anwendungsressourcen bereit und konfigurieren Alarme basierend auf beliebigen Leistungseinbrüchen mithilfe von Amazon CloudWatch oder Drittanbieter-Tools.
☐-☐-☐	Unser Konzept umgeht große Anzahlen von Regeln, die für die Sicherheitsgruppen gelten, die unseren Anwendungs-Instances zugeordnet sind. Durch eine große Anzahl gültiger Regeln für eine Sicherheitsgruppe kann die Leistung eingeschränkt werden.

Ja-Nein-N/V	Kostenoptimierung
☐-☐-☐	Wir wissen, dass ein Infrastrukturereignis möglicherweise mit der zusätzlichen Bereitstellung von Kapazität einhergeht, die nach Beendigung des Ereignisses bereinigt werden muss, um unnötige Kosten zu vermeiden.
☐-☐-☐	Wir verwenden die richtige Größe für alle unsere Infrastrukturkomponenten, einschließlich EC2 Instance-Größe, RDS-DB-Instance-Größe, Caching-Cluster-Knotengröße und -anzahl, Redshift-Cluster-Knotengröße und -anzahl und EBS-Volume-Größe.
☐-☐-☐	Wir verwenden bei Bedarf Spot-Instances. Spot-Instances sind ideal für Workloads mit flexiblen Start- und Endzeiten. Typische Anwendungsfälle für Spot-Instances sind: Batch-Verarbeitung, Berichtserstellung und High-Performance-Computing-Workloads.
☐-☐-☐	Wir haben vorhersehbare Mindestanforderungen an Anwendungskapazitäten und nutzen Reserved Instances. . Reserved Instances ermöglichen die Reservierung von Amazon EC2-Rechenkapazitäten gegen einen wesentlich günstigeren Stundenpreis im Vergleich zu On-Demand-Instance-Preisen.

Notes

[1] https://aws.amazon.com/answers/account-management/aws-tagging-strategies/

[2] https://aws.amazon.com/blogs/aws/resource-groups-and-tagging/

[3] https://aws.amazon.com/sqs/

[4] http://docs.aws.amazon.com/general/latest/gr/rande.html

[5] https://aws.amazon.com/emr/

[6] https://aws.amazon.com/rds/

[7] https://aws.amazon.com/ecs/

[8] https://aws.amazon.com/sns/

[9] https://aws.amazon.com/blogs/compute/using-aws-lambda-with-auto-scaling-lifecycle-hooks/

[10] http://docs.aws.amazon.com/lambda/latest/dg/welcome.html

[11] https://aws.amazon.com/blogs/aws/new-auto-recovery-for-amazon-ec2/

[12] https://aws.amazon.com/answers/configuration-management/aws-infrastructure-configuration-management/

[13] https://d0.awsstatic.com/whitepapers/Big_Data_Analytics_Options_on_AWS%20.pdf

[14] http://docs.aws.amazon.com/Route53/latest/DeveloperGuide/routing-policy.html#routing-policy-latency

[15] https://aws.amazon.com/elasticache/

[16] https://aws.amazon.com/cloudfront/

[17] http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts-on-demand-reserved-instances.html

[18] http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-spot-instances.html

[19] https://aws.amazon.com/about-aws/whats-new/2014/07/31/aws-trusted-advisor-security-and-service-limits-checks-now-free/

[20] https://docs.aws.amazon.com/general/latest/gr/aws_service_limits.html

[21] http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/monitoring_automated_manual.html

[22] http://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/CloudWatch_Dashboards.html

[23] http://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/publishingMetrics.html

[24] https://aws.amazon.com/blogs/aws/new-whitepaper-use-aws-for-disaster-recovery/

[25] http://media.amazonwebservices.com/AWS_Operational_Checklists.pdf

[26] http://d0.awsstatic.com/whitepapers/architecture/AWS_Well-Architected_Framework.pdf

[27] https://aws.amazon.com/premiumsupport/iem/