

AWS Well-Architected Framework

# Säule „Nachhaltigkeit“



# Säule „Nachhaltigkeit“: AWS Well-Architected Framework

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

---

# Table of Contents

Überblick und Einführung .....	i
Einführung .....	1
Cloud Nachhaltigkeit .....	2
Das Modell der geteilten Verantwortlichkeit: .....	3
Nachhaltigkeit der Cloud .....	4
Nachhaltigkeit in der Cloud .....	4
Nachhaltigkeit durch die Cloud: .....	4
Designprinzipien für Nachhaltigkeit in der Cloud: .....	5
Prozess der Verbesserung .....	7
Beispielszenario: .....	8
Identifizieren Sie Ziele für Verbesserungen: .....	8
Ressourcen .....	9
Bewerten Sie spezifische Verbesserungen: .....	9
Proxy-Metriken .....	9
Geschäftsmetriken .....	10
Wichtige Leistungskennzahlen .....	10
Schätzen Sie die Verbesserung .....	11
Bewerten Sie die Verbesserungen .....	12
Setzen Sie Prioritäten und planen Sie Verbesserungen: .....	13
Testen und validieren Sie Verbesserungen: .....	14
Stellen Sie Änderungen für die Produktion bereit. ....	16
Messen Sie die Ergebnisse und wiederholen Sie die Erfolge: .....	16
Nachhaltigkeit als nicht-funktionale Anforderung .....	19
Bewährte Methoden in Bezug auf Nachhaltigkeit in der Cloud .....	21
Auswahl von Regionen .....	21
SUS01-BP01 Auswählen der Region auf Grundlage von Unternehmensanforderungen und Nachhaltigkeitszielen .....	21
Ausrichtung am Bedarf .....	23
SUS02-BP01 Dynamisches Skalieren der Workload-Infrastruktur .....	24
SUS02-BP02 Ausrichten von SLAs an Nachhaltigkeitszielen .....	27
SUS02-BP03 Beenden der Erstellung und Wartung nicht verwendeter Komponenten .....	29
SUS02-BP04 Optimieren der geografischen Platzierung von Workloads auf der Grundlage ihrer Netzwerkanforderungen .....	30

SUS02-BP05 Optimieren von Ressourcen für Teammitglieder im Hinblick auf die ausgeführten Aktivitäten .....	34
SUS02-BP06 Implementierung von Pufferung oder Drosselung, um die Bedarfskurve zu verflachen .....	35
Software und Architektur .....	38
SUS03-BP01 Optimieren von Software und Architektur für asynchrone und geplante Aufträge .....	38
SUS03-BP02 Entfernen oder Refaktorisieren von Workload-Komponenten mit geringer oder keiner Nutzung .....	41
SUS03-BP03 Optimieren von Codebereichen, die die meiste Zeit oder die meisten Ressourcen verbrauchen .....	43
SUS03-BP04 Optimieren der Auswirkungen auf Geräte und Ausrüstung von Kunden .....	45
SUS03-BP05 Verwenden von Softwaremustern und Architekturen, die Datenzugriffs- und Speichermuster optimal unterstützen .....	47
Datenverwaltung .....	49
SUS04-BP01 Implementieren einer Richtlinie für die Klassifizierung von Daten .....	49
SUS04-BP02 Verwenden von Technologien, die Datenzugriff und Speichermuster unterstützen .....	51
SUS04-BP03 Verwalten des Lebenszyklus von Datensätzen mithilfe von Richtlinien .....	56
SUS04-BP04 Verwendung von Elastizität und Automatisierung zur Erweiterung des Block-Speichers oder des Dateisystems .....	59
SUS04-BP05 Entfernen nicht benötigter oder redundanter Daten .....	61
SUS04-BP06 Verwenden geteilter Dateisysteme oder Objektspeicher für den Zugriff auf allgemeine Daten .....	63
SUS04-BP07 Minimieren von Datenübertragungen zwischen Netzwerken .....	66
SUS04-BP08 Sichern von Daten nur in dem Fall, wenn ihre erneute Erstellung schwierig ist .....	68
Hardware und Services .....	70
SUS05-BP01 Verwenden der geringstmöglichen Menge an Hardware zur Erfüllung Ihrer Anforderungen .....	71
SUS05-BP02 Verwenden von Instance-Typen mit den geringsten Auswirkungen .....	73
SUS05-BP03 Verwenden verwalteter Services .....	76
SUS05-BP04 Optimieren der Nutzung von hardwarebasierten Computing-Beschleunigern ....	79
Prozess und Kultur .....	81
SUS06-BP01 Einführen von Methoden, die schnelle Verbesserungen für die Nachhaltigkeit ermöglichen .....	81

---

SUS06-BP02 Konstantes Aktualisieren Ihres Workloads .....	83
SUS06-BP03 Höhere Auslastung von Entwicklungsumgebungen .....	85
SUS06-BP04 Verwenden verwalteter Gerätefarmen für Tests .....	87
Fazit .....	90
Mitwirkende .....	91
Weitere Informationen .....	92
Dokumentversionen .....	93
Hinweise .....	94
AWS Glossary .....	95

# Säule „Nachhaltigkeit“ – AWS-Well-Architected-Framework

Veröffentlichungsdatum: 3. Oktober 2023 ([Dokumentversionen](#))

In diesem Whitepaper geht es speziell um die Säule „Nachhaltigkeit“ des Well-Architected Framework von Amazon Web Services (AWS). Es enthält Designprinzipien, Betriebsanleitungen, bewährte Methoden, potenzielle Kompromisse und Verbesserungspläne, die Sie nutzen können, um die Nachhaltigkeitsziele für Ihre AWS-Workloads zu erreichen.

## Einführung

Das AWS-Well-Architected-Framework unterstützt Sie dabei, die Vor- und Nachteile der Entscheidungen nachzuvollziehen, die Sie beim Erstellen von Workloads in AWS treffen. Das Framework hilft Ihnen, bewährte Architekturmethoden für den Entwurf und Betrieb zuverlässiger, sicherer, effizienter, kostengünstiger und nachhaltiger Workloads in der AWS Cloud zu ermitteln. Es bietet Ihnen die Möglichkeit, Ihre Architekturen konsistent auf die Einhaltung bewährter Methoden zu prüfen und Verbesserungspotenzial zu identifizieren. Mit Well-Architected-Workloads können Sie Ihre Geschäftsergebnisse wesentlich besser unterstützen.

Das Framework basiert auf den folgenden sechs Säulen:

- Operative Exzellenz
- Sicherheit
- Zuverlässigkeit
- Leistungseffizienz
- Kostenoptimierung
- Nachhaltigkeit

Dieses Dokument konzentriert sich auf die Säule der Nachhaltigkeit und dabei insbesondere auf ökologische Nachhaltigkeit. Es richtet sich an Nutzer in technologischen Rollen, z. B. CTOs (Chief Technology Officers), Architekten, Entwickler und Mitglieder von Betriebsteams.

Nach der Lektüre dieses Dokuments kennen Sie die aktuellen AWS-Empfehlungen und -Strategien, die Sie bei der Entwicklung von Cloud-Architekturen mit Blick auf die Nachhaltigkeit anwenden können. Wenn Sie die in diesem Dokument beschriebenen Praktiken anwenden, können Sie Architekturen aufbauen, die die Effizienz maximieren und Verschwendung reduzieren.

# Cloud Nachhaltigkeit

Beim Thema Nachhaltigkeit geht es um die langfristigen ökologischen, wirtschaftlichen und gesellschaftlichen Auswirkungen Ihrer geschäftlichen Aktivitäten. Das [United Nations World Commission on Environment and Development \(UN-Weltkommission für Umwelt und Entwicklung\)](#) definiert nachhaltige Entwicklung als „Entwicklung, die die Bedürfnisse der Gegenwart befriedigt, ohne die Möglichkeiten künftiger Generationen zu gefährden, ihre eigenen Bedürfnisse zu befriedigen“. Die Tätigkeit Ihres Unternehmens oder Ihrer Organisation kann negative Auswirkungen auf die Umwelt haben, wie etwa direkte oder indirekte Kohlendioxidemissionen, nicht-recyclebare Abfälle oder Beeinträchtigungen gemeinsam genutzter Ressourcen, etwa sauberen Wassers.

Bei der Erstellung von Cloud-Workloads bestehen nachhaltige Praktiken darin, die Auswirkungen der genutzten Services zu verstehen, diese über den gesamten Workload-Lebenszyklus hinweg zu quantifizieren, sowie konzeptionelle Grundsätze und bewährte Methoden einzusetzen, die dabei helfen, diese Auswirkungen zu reduzieren. Bei diesem Dokument geht es um Auswirkungen auf die Umwelt, insbesondere um Energieverbrauch und -effizienz, da diese wichtige Faktoren für Architekten sind, die ihre direkten Aktionen zur Reduzierung des Ressourcenverbrauchs beeinflussen.

Wenn Sie sich auf die Umweltauswirkungen konzentrieren, sollten Sie verstehen, wie diese Auswirkungen typischerweise verbucht werden und welche Folgen dies für die Emissionsbilanzierung Ihres Unternehmens hat. Das [Greenhouse Gas Protocol](#) gliedert die Kohlenstoffemissionen in die folgenden Bereiche, zusammen mit relevanten Emissionsbeispielen innerhalb jedes Bereichs für einen Cloud-Anbieter wie AWS:

- Umfang 1: Alle direkten Emissionen aus den Aktivitäten eines Unternehmens oder unter deren Kontrolle. Zum Beispiel die Verbrennung von Treibstoff in Notstromgeneratoren von Rechenzentren.
- Umfang 2: Indirekte Emissionen durch Strom, der für den Betrieb von Rechenzentren und anderen Einrichtungen gekauft und verwendet wird. Zum Beispiel die Emissionen aus der kommerziellen Stromerzeugung.
- Umfang 3: Alle anderen indirekten Emissionen aus Aktivitäten eines Unternehmens aus Quellen, die es nicht kontrollieren kann. AWS Beispiele sind Emissionen im Zusammenhang mit dem Bau von Rechenzentren sowie der Herstellung und dem Transport von IT-Hardware, die in Rechenzentren eingesetzt wird.

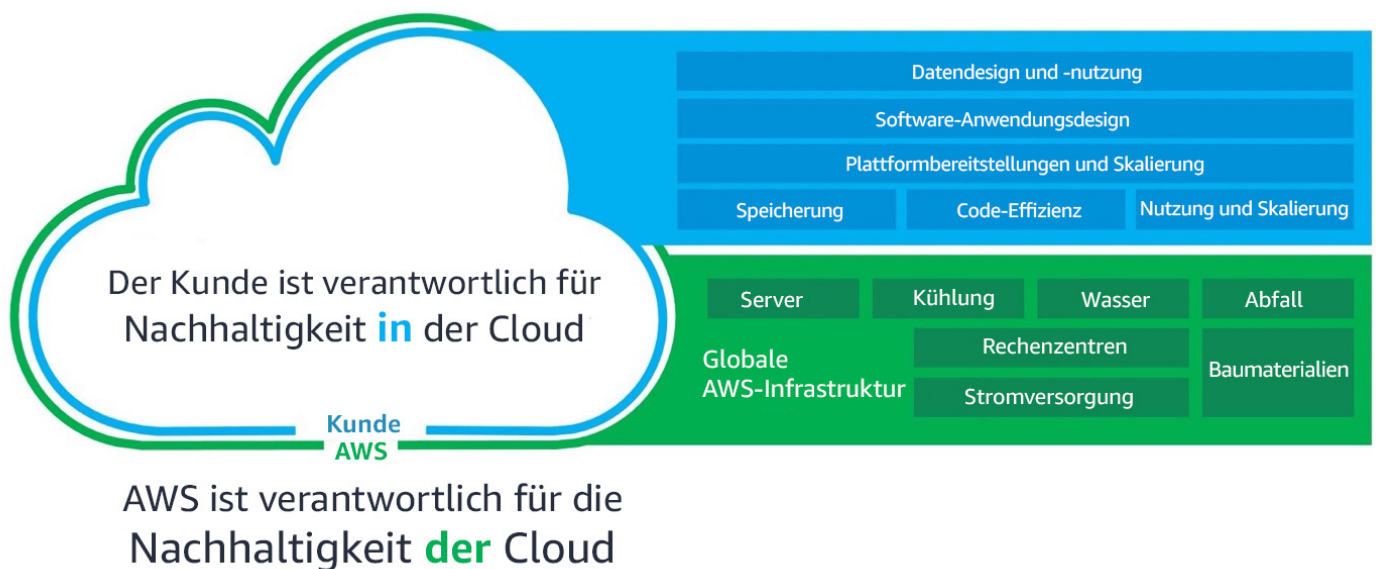
Aus der AWS-Kundenperspektive werden die Emissionen, die durch Ihre auf AWS laufenden Workloads entstehen, als indirekte Emissionen verbucht und sind Teil Ihrer Umfang-3-Emissionen. Jeder eingesetzte Workload erzeugt einen Bruchteil der AWS-Gesamtemissionen aus jedem der vorherigen Bereiche. Der tatsächliche Betrag variiert je nach Workload und hängt von mehreren Faktoren ab, darunter die in Anspruch genommenen AWS-Services, der Energieverbrauch dieser Services, die CO2-Intensität der Stromnetze, die die AWS-Rechenzentren versorgen, in denen sie betrieben werden, und die AWS Beschaffung erneuerbarer Energie.

In diesem Dokument wird zunächst ein Modell der gemeinsamen Verantwortung für ökologische Nachhaltigkeit beschrieben. Anschließend werden bewährte Architekturverfahren vorgestellt, mit denen Sie die Auswirkungen Ihrer Workloads minimieren können, indem Sie die Gesamtressourcen, die für den Betrieb in AWS-Rechenzentren erforderlich sind, reduzieren.

## Das Modell der geteilten Verantwortlichkeit:

Die ökologische Nachhaltigkeit ist eine gemeinsame Verantwortung von Kunden und AWS.

- AWS ist verantwortlich für die Optimierung der Nachhaltigkeit des Cloud-Faktors – die Bereitstellung effizienter, gemeinsam genutzter Infrastrukturen, der verantwortungsvolle Umgang mit Wasser und die Beschaffung erneuerbarer Energie.
- Kunden sind verantwortlich für Nachhaltigkeit in der Cloud – Optimierung von Workloads und Ressourcennutzung und Minimierung der Gesamtressourcen, die für Ihre Workloads bereitgestellt werden müssen.





Modell der geteilten Verantwortung:

## Nachhaltigkeit der Cloud

Cloud-Anbieter haben einen geringeren CO<sub>2</sub>-Fußabdruck und sind energieeffizienter als die typischen On-Premises-Alternativen, da sie in effiziente Strom- und Kühltechnologie investieren, energieeffiziente Serverpopulationen betreiben und hohe Serverauslastungsraten erzielen. Cloud-Workloads reduzieren die Auswirkungen, indem sie gemeinsame Ressourcen wie Netzwerke, Strom, Kühlung und physische Einrichtungen nutzen. Sie können Ihre Cloud-Workloads zu effizienteren Technologien migrieren, sobald diese verfügbar werden, und Cloud-basierte Services verwenden, um Ihre Workloads nachhaltiger auszuführen.

### Ressourcen

- [Die Möglichkeit der CO<sub>2</sub>-Reduzierung durch die Umstellung auf Amazon Web Services.](#)
- [AWS unterstützt nachhaltige Lösungen.](#)

## Nachhaltigkeit in der Cloud

Nachhaltigkeit in der Cloud ist ein kontinuierliches Bestreben, das sich in erster Linie auf die Reduzierung des Energieverbrauchs und die Effizienz aller Komponenten eines Workloads konzentriert. Dazu muss der maximale Nutzen aus den bereitgestellten Ressourcen gezogen und die insgesamt erforderlichen Ressourcen müssen minimiert werden. Diese Bemühung kann von der anfänglichen Auswahl einer effizienten Programmiersprache, der Einführung moderner Algorithmen, der Nutzung effizienter Datenspeichertechniken, der Bereitstellung einer korrekt dimensionierten und effizienten Recheninfrastruktur bis hin zur Minimierung der Anforderungen an leistungsstarke Endbenutzerhardware reichen.

## Nachhaltigkeit durch die Cloud:

Sie können nicht nur die Auswirkungen der von Ihnen bereitgestellten Workloads minimieren, sondern auch AWS Cloud-Workloads ausführen, die zur Unterstützung Ihrer umfassenderen Nachhaltigkeitsprobleme entwickelt wurden. Beispiele für diese Herausforderungen sind die Verringerung der CO<sub>2</sub>-Emissionen, die Senkung des Energieverbrauchs, das Recycling von Wasser oder die Reduzierung von Abfall in anderen Bereichen Ihres Unternehmens oder Ihrer Organisation.

Nachhaltigkeit durch die Cloud bedeutet, dass Sie die AWS-Technologie nutzen, um ein breiteres Nachhaltigkeitsproblem zu lösen. Sie können zum Beispiel einen Machine Learning Service wie

[Amazon Monitron](#) nutzen, um abnormales Verhalten in Industriemaschinen zu erkennen. Anhand dieser Erkennungsdaten können Sie eine vorbeugende Wartung durchführen, um das Risiko von Umweltschäden durch unerwartete Geräteausfälle zu verringern und sicherzustellen, dass die Maschinen weiterhin mit höchster Effizienz arbeiten.

## Designprinzipien für Nachhaltigkeit in der Cloud:

Wenden Sie diese Designprinzipien an, wenn Sie Ihre Cloud-Workloads entwickeln, um die Nachhaltigkeit zu maximieren und die Auswirkungen zu minimieren.

- **Verstehen Sie Ihre Auswirkungen:** Messen Sie die Auswirkungen Ihrer Cloud-Workloads und modellieren Sie diese Auswirkungen für die Zukunft. berücksichtigen Sie dabei alle relevanten Faktoren, darunter Auswirkungen durch die Verwendung Ihrer Produkte durch Kunden sowie solche durch deren Außerbetriebnahme und Entsorgung. Vergleichen Sie den produktiven Output mit den Gesamtauswirkungen Ihrer Cloud-Workloads, indem Sie die für jede Arbeitseinheit erforderlichen Ressourcen und die damit verbundenen Emissionen ermitteln. Anhand dieser Daten können Sie Leistungskennzahlen (KPIs) einrichten, Möglichkeiten zur Verbesserung der Produktivität bei gleichzeitiger Reduzierung der Auswirkungen finden und berechnen, wie sich vorgeschlagene Änderungen im Zeitverlauf auswirken werden.
- **Legen Sie Nachhaltigkeitsziele fest:** Formulieren Sie für alle Cloud-Workloads langfristige Nachhaltigkeitsziele wie etwa die Reduzierung der pro Transaktion erforderlichen Computing- und Speicherressourcen. Modellieren Sie den ROI von Verbesserungen in Bezug auf die Nachhaltigkeit vorhandener Workloads. Stellen Sie den Besitzern die nötigen Ressourcen zur Verfügung, um in Nachhaltigkeitsziele investieren zu können. Planen Sie wachstumsorientiert und gestalten Sie Ihre Workloads so, dass das Wachstum mit geringeren Auswirkungen einhergeht – gemessen in einer sinnvollen Einheit, etwa pro Benutzer oder pro Transaktion. Ziele helfen Ihnen, die allgemeinen Nachhaltigkeitsziele Ihres Unternehmens oder Ihrer Organisation zu erreichen, Rückschritte zu identifizieren und Bereiche mit Verbesserungsmöglichkeiten zu priorisieren.
- **Maximieren Sie Ihre Auslastung:** Sorgen Sie für Workloads angemessenen Umfangs und nutzen Sie effiziente Designprinzipien, um hohe Auslastung zu gewährleisten und die Energieeffizienz der zugrunde liegenden Hardware so zu maximieren. Zwei Hosts mit 30 % Auslastung sind aufgrund des grundlegenden Energieverbrauchs pro Host weniger effizient als ein Host mit 60 % Auslastung. Gleichzeitig sollten Sie nicht genutzte Ressourcen, Verarbeitungsvorgänge und Speicher beseitigen oder minimieren, um den Gesamtenergieverbrauch für Ihren Workload zu senken.

- **Antizipieren und nutzen Sie neue und effizientere Hardware- und Software-Angebote:** Unterstützen Sie die Verbesserungen, die Ihre Partner und Lieferanten in früheren Prozessphasen vornehmen, um die Auswirkungen Ihrer Cloud-Workloads zu reduzieren. Achten Sie stets auf neue und effizientere Hardware- und Software-Angebote. Planen Sie für Flexibilität, damit neue effiziente Technologien schnell eingeführt werden können.
- **Verwenden Sie verwaltete Services:** Die gemeinsame Nutzung von Services über eine breite Kundenbasis hinweg hilft dabei, die Ressourcennutzung zu maximieren und dadurch den Umfang der Infrastruktur zu verringern, der für die Unterstützung Ihrer Cloud-Workloads erforderlich ist. So können Kunden die Auswirkungen allgemeiner Rechenzentrumskomponenten wie Energieversorgung und Netzwerk teilen, indem sie Workloads zur AWS Cloud migrieren und verwaltete Services einführen, z. B. AWS Fargate für Serverless-Container. Dabei kann AWS skalierbar ausgeführt werden und ist für einen effizienten Betrieb verantwortlich. Verwenden Sie verwaltete Services, die dabei helfen können, Ihre Auswirkungen zu verringern, wie etwa die automatische Verschiebung selten genutzter Daten in „kalte“ Speicher mit Amazon S3-Lifecycle-Konfigurationen oder Amazon EC2 Auto Scaling, um Ihre Kapazitäten an die jeweiligen Anforderungen anzupassen.
- **Reduzieren Sie nachgelagerte Auswirkungen Ihrer Cloud-Workloads:** Senken Sie den Energie- oder Ressourcenverbrauch für die Nutzung Ihrer Services. Reduzieren oder beseitigen Sie die Erfordernis für Kunden, ihre Geräte zu aktualisieren, um Ihre Services nutzen zu können. Verwenden Sie in Ihren Tests Gerätefarmen, um die zu erwartenden Auswirkungen zu verstehen, und führen Sie Tests mit Kunden durch, um die tatsächlichen Auswirkungen der Nutzung Ihrer Services zu erkennen.

# Prozess der Verbesserung

Zur Verbesserung der Architektur gehört es, zu verstehen, was Sie haben und was Sie tun können, um Verbesserungen zu erzielen. Außerdem sollten Sie Zielbereiche für Verbesserungen auswählen, Verbesserungsmöglichkeiten testen, jene übernehmen, die sich dabei als erfolgreich erweisen, Ihren Erfolg quantifizieren und das weitergeben, was Sie gelernt haben, damit dies anderswo repliziert werden kann. Anschließend beginnt dieser Zyklus von neuem.

Zu den Zielen für Verbesserungen kann Folgendes gehören:

- Beseitigung von Verschwendung, geringer Nutzung und ungenutzten oder ungenutzten Ressourcen
- Maximierung des Wertes der verbrauchten Ressourcen

## Note

Nutzen Sie alle Ihnen zur Verfügung stehenden Ressourcen und erledigen Sie die gleiche Arbeit mit einem Minimum an Ressourcen.

Beseitigen Sie in den frühen Phasen der Optimierung zunächst Bereiche mit Verschwendung oder geringer Auslastung und gehen Sie dann zu gezielteren Optimierungen über, die Ihrem spezifischen Workload entsprechen.

Überwachen Sie die Auswirkungen auf den Ressourcenverbrauch im Zeitverlauf. Ermitteln Sie, wo akkumulierte Änderungen zu ineffizientem Ressourcenverbrauch oder zu deutlichen Anstiegen des Verbrauchs führen. Stellen Sie fest, welche Verbesserungen im Hinblick auf diese Änderungen beim Ressourcenverbrauch erforderlich sind, und implementieren Sie die von Ihnen priorisierten Verbesserungen.

Die folgenden Schritte sind als iterativer Prozess konzipiert, der auf Nachhaltigkeit ausgerichtete Verbesserungen für Cloud-Workloads evaluiert, priorisiert, testet und implementiert.

1. Identifizieren Sie Ziele für Verbesserungen: Überprüfen Sie Ihre Workloads anhand der in diesem Dokument genannten bewährten Methoden für Nachhaltigkeit und ermitteln Sie Ziele für Verbesserungen.

2. Bewerten Sie spezifische Verbesserungen: Bewerten Sie ausgewählte Änderungen hinsichtlich potenzieller Verbesserungen, geplanter Kosten und Geschäftsrisiken.
3. Setzen Sie Prioritäten und planen Sie Verbesserungen: Setzen Sie Prioritäten für Änderungen, die die größten Verbesserungen bei geringsten Kosten und Risiken bieten, und erstellen Sie einen Plan für Tests und Implementierung.
4. Testen und validieren Sie Verbesserungen: Implementieren Sie Änderungen in Testumgebungen, um deren Verbesserungspotenzial zu validieren.
5. Stellen Sie Änderungen für die Produktion bereit: Implementieren Sie Änderungen in Produktionsumgebungen.
6. Messen Sie die Ergebnisse und wiederholen Sie die Erfolge: Suchen Sie nach Möglichkeiten, Erfolge in allen Workloads zu wiederholen, und machen Sie Änderungen rückgängig, die zu inakzeptablen Ergebnissen geführt haben.

## Beispielszenario:

Das folgende Beispielszenario wird später in diesem Dokument zur Veranschaulichung der einzelnen Schritte des Verbesserungsprozesses herangezogen.

Ihr Unternehmen hat einen Workload, der komplexe Image-Manipulationen auf Amazon EC2-Instances durchführt und die geänderten und ursprünglichen Dateien für den Benutzerzugriff speichert. Die Verarbeitungsaktivitäten sind rechenintensiv und die Ausgabedateien sind extrem groß.

## Identifizieren Sie Ziele für Verbesserungen:

Lernen Sie die bewährten Methoden kennen, mit denen Sie Ihre Nachhaltigkeitsziele erreichen können. Ausführliche Beschreibungen dieser [bewährten Methoden](#) und Empfehlungen für Verbesserungen finden Sie weiter unten in diesem Dokument.

Überprüfen Sie Ihre Workloads und die verwendeten Ressourcen. Identifizieren Sie kritische Punkte, wie z. B. große Bereitstellungen und häufig genutzte Ressourcen. Untersuchen Sie diese kritischen Punkte auf Möglichkeiten zur Verbesserung der effektiven Nutzung Ihrer Ressourcen und zur Verringerung der Gesamtressourcen, die zum Erreichen Ihrer Geschäftsergebnisse erforderlich sind.

Überprüfen Sie Ihren Workload anhand bewährter Methoden und ermitteln Sie Verbesserungsmöglichkeiten.

Indem Sie diesen Schritt anwenden, [Beispielszenario](#): haben Sie die folgenden bewährten Methoden als mögliche Ziele für Verbesserungen identifiziert:

- Verwenden der geringstmöglichen Menge an Hardware zur Erfüllung Ihrer Anforderungen
- Technologien verwenden, die Ihre Datenzugriffs- und Speichermuster am besten unterstützen

## Ressourcen

- [Optimieren Ihrer AWS-Infrastruktur für Nachhaltigkeit, Teil I: Datenverarbeitung](#)
- [Optimieren Ihrer AWS-Infrastruktur für Nachhaltigkeit, Teil II: Speicher](#)
- [Optimieren Ihrer AWS-Infrastruktur für Nachhaltigkeit, Teil III: Netzwerke](#)

## Bewerten Sie spezifische Verbesserungen:

Verstehen Sie die Ressourcen, die Ihr Workload für die Durchführung einer Arbeitseinheit vorsieht. Bewerten Sie potenzielle Verbesserungen und schätzen Sie ihre möglichen Auswirkungen, die Kosten für die Implementierung und die damit verbundenen Risiken ab.

Um Verbesserungen im Laufe der Zeit zu messen, müssen Sie zunächst verstehen, was Sie in AWS bereitgestellt haben und wie diese Ressourcen verbraucht werden.

Beginnen Sie mit einem vollständigen Überblick über Ihren AWS-Verbrauch und nutzen Sie AWS-Kosten- und Verbrauchsberichte, um kritische Punkte zu identifizieren. Verwenden Sie diesen [AWS-Beispielcode](#), um Sie bei der Überprüfung und Analyse Ihres Berichts mit Hilfe von Amazon Athena zu unterstützen.

## Proxy-Metriken

Wenn Sie bestimmte Änderungen bewerten, müssen Sie auch beurteilen, welche Metriken die Auswirkungen dieser Änderung auf die zugehörige Ressource am besten quantifizieren. Diese Metriken werden Proxy-Metriken genannt. Wählen Sie Proxy-Metriken, die die Art der Verbesserung, die Sie bewerten, und die Ressourcen, auf die die Verbesserung abzielt, am besten widerspiegeln. Diese Metriken können sich im Laufe der Zeit weiterentwickeln.

Zu den Ressourcen, die zur Unterstützung Ihres Workloads bereitgestellt werden, gehören Rechen-, Speicher- und Netzwerkressourcen. Bewerten Sie die bereitgestellten Ressourcen anhand Ihrer Proxy-Metriken, um zu sehen, wie diese Ressourcen verbraucht werden.

Verwenden Sie Ihre Proxy-Metriken, um die Ressourcen zu messen, die zur Erreichung der Geschäftsergebnisse bereitgestellt werden.

Ressource	Beispiel-Proxy-Metriken	Verbesserungsziele
Datenverarbeitung	vCPU-Minuten	Maximieren der Nutzung der bereitgestellten Ressourcen
Speicherung	Bereitgestellte GB	Reduzieren der gesamten Bereitstellung
Netzwerk	Übertragene GB oder Pakete	Reduzieren der übertragenen Gesamtmenge und der übertragenen Entfernung

## Geschäftsmetriken

Wählen Sie Geschäftsmetriken aus, um das Erreichen von Geschäftsergebnissen zu quantifizieren. Ihre Geschäftsmetriken sollten den Wert widerspiegeln, den Ihr Workload bietet, z. B. die Anzahl der gleichzeitig aktiven Benutzer, die Anzahl der API-Aufrufe oder die Anzahl der abgeschlossenen Transaktionen. Diese Metriken können sich im Laufe der Zeit weiterentwickeln. Seien Sie vorsichtig bei der Bewertung von finanzbasierten Geschäftsmetriken, da Unstimmigkeiten beim Wert von Transaktionen Vergleiche ungültig machen.

## Wichtige Leistungskennzahlen

Teilen Sie die bereitgestellten Ressourcen durch die erzielten Geschäftsergebnisse, um die bereitgestellten Ressourcen pro Arbeitseinheit zu ermitteln.

$$\text{Ressourcen pro Arbeitseinheit bereitgestellt} = \frac{\text{Proxy-Metrik für bereitgestellte Ressource}}{\text{Geschäftsmetrik für Ergebnis}}$$

### KPI-Formel

Verwenden Sie Ihre Ressourcen pro Arbeitseinheit als Ihre KPIs. Festlegung von Basiswerten auf der Grundlage der bereitgestellten Ressourcen als Grundlage für Vergleiche.

Ressource	Beispiel-KPIs	Verbesserungsziele
Datenverarbeitung	vCPU-Minuten pro Transaktion	Maximieren der Nutzung der bereitgestellten Ressourcen
Speicherung	GB pro Transaktion	Reduzieren der gesamten Bereitstellung
Netzwerk	Übertragene GB pro Transaktion oder übertragene Pakete pro Transaktion	Reduzieren der übertragenen Gesamtmenge und der übertragenen Entfernung

## Schätzen Sie die Verbesserung

Schätzen Sie die Verbesserung sowohl als quantitative Verringerung der bereitgestellten Ressourcen (wie durch Ihre Proxy-Kennzahlen angegeben) als auch als prozentuale Veränderung im Vergleich zu Ihren Basisressourcen, die pro Arbeitseinheit bereitgestellt werden.

Ressource	Beispiel-KPIs	Verbesserungsziele
Datenverarbeitung	Prozentuale Reduzierung der vCPU-Minuten pro Transaktion	Maximieren Sie Ihre Auslastung
Speicherung	Prozentuale Reduzierung der GB pro Transaktion	Reduzieren der gesamten Bereitstellung
Netzwerk	Prozentuale Reduzierung der pro Transaktion übertragenen GB oder Pakete pro Transaktion	Reduzieren der übertragenen Gesamtmenge und der übertragenen Entfernung



## Bewerten Sie die Verbesserungen

Bewerten Sie mögliche Verbesserungen anhand des erwarteten Nettonutzens. Bewerten Sie den Zeit-, Kosten- und Arbeitsaufwand für die Implementierung und Wartung sowie die Geschäftsrisiken, z. B. unvorhergesehene Auswirkungen.

Gezielte Verbesserungen stellen oft einen Kompromiss zwischen den verschiedenen Arten von verbrauchten Ressourcen dar. So können Sie z. B. ein Ergebnis speichern, um den Rechenaufwand zu verringern, oder Sie können Daten verarbeiten, bevor Sie das Ergebnis an einen Client senden, um die übertragenen Daten zu begrenzen. Diese [Kompromisse](#) werden später noch ausführlicher erörtert.

Berücksichtigen Sie bei der Bewertung der Risiken für Ihren Workload auch nicht-funktionale Anforderungen wie Sicherheit, Zuverlässigkeit, Leistungseffizienz, Kostenoptimierung und die Auswirkungen von Verbesserungen auf Ihre Fähigkeit, Ihren Workload zu betreiben.

Indem Sie diesen Schritt anwenden, [Beispielszenario](#): bewerten Sie die angestrebten Verbesserungen mit den folgenden Ergebnissen:

Bewährte Methode	Gezielte Verbesserung	Potential	Kosten	Risiko
Verwenden der geringstmöglichen Menge an Hardware zur Erfüllung Ihrer Anforderungen	Implementierung einer vorausschauenden Skalierung zur Verringerung von Zeiten geringer Auslastung	Mittel	Niedrig	Niedrig
Technologien verwenden, die Ihre Datenzugriffs- und Speichermuster am besten unterstützen	Implementierung effektiver Komprimierungsmechanismen, um die Gesamtspeicherung und die dafür benötigte	Hoch	Niedrig	Niedrig

Bewährte Methode	Gezielte Verbesserung	Potential	Kosten	Risiko
	Zeit zu reduzieren			

Die Implementierung von prädikativer Planung (Predictive Scheduling) reduziert die vCPU-Stunden, die von nicht oder nur unzureichend ausgelasteten Instances verbraucht werden, und bietet mit einer geschätzten Reduzierung der verbrauchten Ressourcen um 11 % moderate Vorteile gegenüber bestehenden Skalierungsmechanismen. Die damit verbundenen Kosten sind gering und umfassen die Konfiguration der Cloud-Ressourcen und den Betrieb der vorausschauenden Skalierung für Amazon EC2 Auto Scaling. Das Risiko ist eine eingeschränkte Leistung, wenn die Skalierung reaktiv erfolgt, wenn die Nachfrage die Prognosen übersteigt.

Die Implementierung einer effektiveren Komprimierung wird sich erheblich auf die Dateigröße aller Original- und bearbeiteten Bilder auswirken und den Speicherbedarf in der Produktion um schätzungsweise 25 % verringern. Die Implementierung des neuen Algorithmus ist eine Substitution mit geringem Aufwand und Risiko.

## Setzen Sie Prioritäten und planen Sie Verbesserungen:

Legen Sie die Prioritäten für die ermittelten Verbesserungen auf der Grundlage der größten zu erwartenden Auswirkungen mit den geringsten Kosten und akzeptablen Risiken fest.

Entscheiden Sie, auf welche Verbesserungen Sie sich zunächst konzentrieren wollen, und nehmen Sie diese in Ihre Ressourcenplanung und Ihre Entwicklungs-Roadmap auf.

Indem Sie diesen Schritt anwenden, [Beispielszenario](#):priorisieren Sie die angestrebten Verbesserungen wie folgt:

Priorität	Verbesserung	Potential	Kosten	Risiko
1	Implementieren wirksamerer Komprimierungsmechanismen	Hoch	Niedrig	Niedrig

Priorität	Verbesserung	Potential	Kosten	Risiko
2	Implementieren der vorausschauenden Skalierung	Mittel	Niedrig	Niedrig

Das hohe Potenzial, die geringen Kosten und das Risiko einer Aktualisierung der Dateikomprimierung machen sie zu einem hochwertigen Ziel für Ihr Unternehmen und zu einer Priorität gegenüber der Implementierung von vorausschauender Skalierung (Predictive Scaling). Sie legen fest, dass die Implementierung der prädiktiven Skalierung mit ihrem mittleren Wirkungspotenzial, den geringen Kosten und dem geringen Risiko die vorrangige Verbesserung nach Abschluss der Dateikomprimierung sein sollte.

Sie beauftragen ein Teammitglied mit der Implementierung einer verbesserten Dateikomprimierung und fügen Ihrem Backlog eine vorausschauende Skalierung hinzu.

## Testen und validieren Sie Verbesserungen:

Führen Sie kleine Tests mit minimalen Investitionen durch, um das Risiko eines groß angelegten Aufwands zu verringern.

Implementieren Sie eine repräsentative Kopie Ihres Workloads in Ihrer Testumgebung, um die Kosten und Risiken für die Durchführung von Tests und Validierungen zu begrenzen. Führen Sie einen vordefinierten Satz von Testtransaktionen durch, messen Sie die bereitgestellten Ressourcen und ermitteln Sie die pro Arbeitseinheit verbrauchten Ressourcen, um eine Test-Baseline zu erstellen.

Implementieren Sie die angestrebte Verbesserung in der Testumgebung und wiederholen Sie den Test mit der gleichen Methodik unter den gleichen Bedingungen. Messen Sie dann die bereitgestellten Ressourcen und die pro Arbeitseinheit verbrauchten Ressourcen, wenn Sie Ihre Verbesserung umgesetzt haben.

Berechnen Sie die prozentuale Veränderung der pro Arbeitseinheit bereitgestellten Ressourcen im Vergleich zum Ausgangswert und bestimmen Sie die erwartete quantitative Reduzierung der in Ihrer Produktionsumgebung bereitgestellten Ressourcen. Vergleichen Sie diese Werte mit den erwarteten Werten. Bestimmen Sie, ob das Ergebnis ein akzeptables Maß an Verbesserung darstellt. Beurteilen Sie, ob der Nettonutzen der Verbesserung aufgrund von Kompromissen bei den zusätzlich verbrauchten Ressourcen inakzeptabel ist.

Bestimmen Sie, ob die Verbesserung ein Erfolg ist und ob Ressourcen in die Umsetzung der Änderung in der Produktion investiert werden sollten. Wenn die Änderung zu diesem Zeitpunkt als nicht erfolgreich bewertet wird, leiten Sie Ihre Ressourcen um, um Ihr nächstes Ziel zu testen und zu validieren, und setzen Sie Ihren Verbesserungszyklus fort.

Prozentuale Reduzierung der bereitgestellten Ressourcen pro Arbeitseinheit	Quantitativer Rückgang der bereitgestellten Ressourcen	Aktion
Erwartungen erfüllt	Erwartungen erfüllt	Fortfahren mit der Verbesserung
Erwartungen nicht erfüllt	Erwartungen erfüllt	Fortfahren mit der Verbesserung
Erwartungen erfüllt	Erwartungen nicht erfüllt	Alternative Verbesserung anstreben
Erwartungen nicht erfüllt	Erwartungen nicht erfüllt	Alternative Verbesserung anstreben

Indem Sie diesen Schritt anwenden, [Beispielszenario](#):führen Sie Tests durch, um den Erfolg zu überprüfen.

Nach den Tests mit dem verbesserten Komprimierungsalgorithmus entsprach die prozentuale Verringerung der pro Arbeitseinheit bereitgestellten Ressourcen (der für das ursprüngliche und das geänderte Bild benötigte Speicherplatz) den Erwartungen, mit einer durchschnittlichen Verringerung des bereitgestellten Speicherplatzes um 30 % und einer vernachlässigbaren Erhöhung der Rechenlast.

Sie stellen fest, dass die zusätzlichen Rechenressourcen, die für die Anwendung des verbesserten Komprimierungsalgorithmus auf bestehende Dateien in der Produktion erforderlich sind, im Vergleich zu der erzielten Speicherplatzeinsparung unbedeutend sind. Sie haben den Erfolg der quantitativen Verringerung der erforderlichen Ressourcen (TB Speicherplatz) bestätigt, und die Verbesserung ist für die Produktionsbereitstellung freigegeben.

## Stellen Sie Änderungen für die Produktion bereit.

Implementieren Sie getestete, validierte und genehmigte Verbesserungen in der Produktion. Führen Sie die Implementierung mit begrenzten Bereitstellungen durch, bestätigen Sie die Funktionalität Ihres Workloads, testen Sie die tatsächliche Verringerung der bereitgestellten Ressourcen und des Ressourcenverbrauchs pro Arbeitseinheit innerhalb der begrenzten Bereitstellung und prüfen Sie auf unbeabsichtigte Folgen der Änderung. Fahren Sie nach erfolgreichen Tests mit der vollständigen Bereitstellung fort.

Machen Sie Änderungen rückgängig, wenn Tests fehlschlagen oder Sie auf inakzeptable, unbeabsichtigte Folgen Ihrer Änderung stoßen.

Indem Sie diesen Schritt anwenden, [Beispielszenario](#): ergreifen Sie die folgenden Maßnahmen.

Sie implementieren die Änderungen in der Produktion mit einer begrenzten Bereitstellung durch eine Blue-Green-Bereitstellungsmethodik. Die Funktionstests für die neu installierten Instances sind erfolgreich. Sie sehen eine durchschnittliche Reduzierung des bereitgestellten Speichers für Original- und manipulierte Bilddateien um 26%. Bei der Komprimierung neuer Dateien gibt es keine Anzeichen für einen Anstieg der Rechenlast.

Sie bemerken eine unerwartete Verringerung der Zeit, die für die Komprimierung von Bilddateien benötigt wird, und führen dies auf den stark optimierten Code für den neuen Komprimierungsalgorithmus zurück.

Sie fahren mit der vollständigen Bereitstellung der neuen Version fort.

## Messen Sie die Ergebnisse und wiederholen Sie die Erfolge:

Messen Sie die Ergebnisse und wiederholen Sie die Erfolge auf folgende Weise:

- Messen Sie die anfängliche Verbesserung der bereitgestellten Ressourcen pro Arbeitseinheit und den quantitativen Rückgang der bereitgestellten Ressourcen.
- Vergleichen Sie erste Schätzungen und Testergebnisse mit Ihren Produktionsmessungen. Ermitteln Sie Faktoren, die zu den Unterschieden beigetragen haben könnten, und aktualisieren Sie gegebenenfalls Ihre Schätz- und Testmethoden.
- Bestimmen Sie den Erfolg und den Grad des Erfolgs und teilen Sie die Ergebnisse mit den Beteiligten.
- Wenn Sie Änderungen aufgrund fehlgeschlagener Tests oder unbeabsichtigter negativer Folgen der Änderung rückgängig machen mussten, ermitteln Sie die dafür verantwortlichen Faktoren.

Iterieren Sie, wo dies möglich ist, oder bewerten Sie neue Ansätze, um die Ziele der Veränderung zu erreichen.

- Nehmen Sie das, was Sie gelernt haben, legen Sie Standards fest und wenden Sie erfolgreiche Verbesserungen auf andere Systeme an, die ebenfalls davon profitieren können. Erfassen und teilen Sie Ihre Methodik, die zugehörigen Artefakte und den Nettonutzen mit anderen Teams und Organisationen, damit andere Ihren Standard übernehmen und Ihren Erfolg nachahmen können.
- Überwachen Sie die bereitgestellten Ressourcen pro Arbeitseinheit und verfolgen Sie Änderungen und Gesamtauswirkungen im Zeitverlauf. Änderungen Ihres Workloads oder der Art und Weise, wie Ihre Kunden Ihr Workload in Anspruch nehmen, können sich auf die Wirksamkeit Ihrer Verbesserungen auswirken. Bewerten Sie die Verbesserungsmöglichkeiten neu, wenn Sie kurzfristig einen erheblichen Rückgang der Effektivität Ihrer Verbesserung oder im Laufe der Zeit einen kumulierten Rückgang der Effektivität feststellen.
- Quantifizieren Sie den Nettonutzen Ihrer Verbesserung im Laufe der Zeit (einschließlich des Nutzens für andere Teams, die Ihre Verbesserung angewandt haben, falls verfügbar), um die Rentabilität Ihrer Verbesserungsaktivitäten aufzuzeigen.

Indem Sie diesen Schritt anwenden, [Beispielszenario](#): messen Sie die folgenden Ergebnisse:

Ihr Workload zeigt eine anfängliche Verbesserung von 23 % bei den Speicheranforderungen, nachdem der neue Komprimierungsalgorithmus auf bestehende Bilddateien angewandt wurde.

Der gemessene Wert stimmt weitgehend mit den anfänglichen Schätzungen (25 %) überein, und der signifikante Unterschied im Vergleich zu den Tests (30 %) ist darauf zurückzuführen, dass die bei den Tests verwendeten Bilddateien nicht repräsentativ für die in der Produktion vorhandenen Bilddateien sind. Sie ändern das Test-Image-Set so, dass es den Bildern in der Produktion besser entspricht.

Die Verbesserung wird als voller Erfolg gewertet. Die Gesamtreduzierung des bereitgestellten Speicherplatzes liegt 2 % unter den geschätzten 25 %, aber 23 % sind immer noch eine enorme Verbesserung der Nachhaltigkeitsauswirkungen und gehen mit einer entsprechenden Kosteneinsparung einher.

Die einzigen unbeabsichtigten Folgen dieser Änderung sind die Verringerung der Zeit, die für die Komprimierung benötigt wird, und eine entsprechende Verringerung des vCPU-Verbrauchs. Diese Verbesserungen sind auf den hoch optimierten Code zurückzuführen.

Sie richten ein internes Open-Source-Projekt ein, in dem Sie Ihren Code, die zugehörigen Artefakte, Anleitungen zur Implementierung der Änderung und die Ergebnisse Ihrer Implementierung bereitstellen. Das interne Open-Source-Projekt macht es Ihren Teams leicht, den Code für alle

Anwendungsfälle der persistenten Dateispeicherung zu übernehmen. Ihre Teams übernehmen die Verbesserung als Standard. Ein weiterer Vorteil des internen Open-Source-Projekts besteht darin, dass jeder, der die Lösung übernimmt, von den Verbesserungen der Lösung profitiert, und jeder kann Verbesserungen zum Projekt beitragen.

Sie veröffentlichen Ihren Erfolg und machen das Open-Source-Projekt in Ihrem Unternehmen bekannt. Jedes Team, das die Lösung übernimmt, repliziert den Nutzen mit minimalen Investitionen und trägt zum Nettonutzen Ihrer Investition bei. Sie veröffentlichen diese Daten als eine fortlaufende Erfolgsgeschichte.

Sie überwachen die Auswirkungen der Verbesserungen im Laufe der Zeit und nehmen bei Bedarf Änderungen an dem internen Open-Source-Projekt vor.

# Nachhaltigkeit als nicht-funktionale Anforderung

Die Aufnahme von Nachhaltigkeit in die Liste der geschäftlichen Anforderungen kann zu kostengünstigeren Lösungen führen. Wenn Sie sich darauf konzentrieren, die von Ihnen genutzten Ressourcen besser zu nutzen und weniger von ihnen zu verbrauchen, führt dies direkt zu Kosteneinsparungen auf AWS, da Sie nur für das bezahlen, was Sie benutzen.

Das Erreichen von Nachhaltigkeitszielen erfordert möglicherweise keine gleichwertigen Kompromisse bei einer oder mehreren anderen traditionellen Messgrößen wie Betriebszeit, Verfügbarkeit oder Reaktionszeit. Sie können erhebliche Nachhaltigkeitsgewinne erzielen, ohne dass dies messbare Auswirkungen auf das Serviceniveau hat. Wenn geringfügige Kompromisse erforderlich sind, können die durch diese Kompromisse erzielten Verbesserungen der Nachhaltigkeit die Veränderung der Servicequalität überwiegen.

Ermutigen Sie Ihre Teammitglieder, bei der Entwicklung funktionaler Anforderungen kontinuierlich mit Verbesserungen der Nachhaltigkeit zu experimentieren. Teams sollten bei der Festlegung von Zielen auch Proxy-Metriken einbeziehen, um sicherzustellen, dass sie die Ressourcenintensität bei der Entwicklung ihres Workload bewerten.

Im Folgenden finden Sie Beispiele für Kompromisse, die den Verbrauch von Cloud-Ressourcen reduzieren können:

**Anpassen der Ergebnisqualität:** Mithilfe von Approximate Computing können Sie bei der Ergebnisqualität Kompromisse eingehen, um die Workload-Intensität zu verringern. Beim Approximate Computing werden Optimierungsmöglichkeiten ermittelt, indem geprüft wird, was produziert wird und was Kunden brauchen. Wenn Sie Ihre Daten zum Beispiel in eine feste Datenstruktur einbringen, können Sie den ORDER BY-Operator in SQL weglassen, um unnötige Verarbeitungen zu vermeiden, Ressourcen zu sparen und trotzdem eine akzeptable Antwort zu erhalten.

**Anpassen der Reaktionszeit:** Eine Antwort mit einer langsameren Reaktionszeit kann CO<sub>2</sub> reduzieren, indem der gemeinsame Overhead minimiert wird. Die Verarbeitung flüchtiger Ad-hoc-Aufgaben kann zu Startup-Overhead führen. Gruppieren und verarbeiten Sie Aufgaben in Stapeln, anstatt bei jeder neuen Aufgabe für den Overhead zu zahlen. Die Stapelverarbeitung führt zu einer verlängerten Reaktionszeit, verringert jedoch den gemeinsamen Overhead für das Hochfahren der Instance, das Herunterladen des Quellcodes und die Ausführung des Prozesses.

**Anpassen der Verfügbarkeit:** Mit AWS können Sie Redundanz hinzufügen und Hochverfügbarkeitsziele mit nur wenigen Klicks erreichen. Sie können die Redundanz durch



Techniken wie statische Stabilität erhöhen, indem Sie ungenutzte Ressourcen bereitstellen, die immer zu einer geringeren Auslastung führen. Bewerten Sie bei der Festlegung von Zielen die Bedürfnisse des Unternehmens. Relativ geringe Abstriche bei der Verfügbarkeit können zu wesentlich größeren Verbesserungen bei der Nutzung führen. Das Architekturmuster der statischen Stabilität beinhaltet beispielsweise die Bereitstellung ungenutzter Failover-Kapazitäten, um die Last nach einem Komponentenfehler sofort zu übernehmen. Eine Lockerung der Verfügbarkeitsanforderung kann den Bedarf an ungenutzten Online-Kapazitäten beseitigen, da die Automatisierung Zeit für die Bereitstellung von Ersatzressourcen bietet. Das Hinzufügen von On-Demand-Failover-Kapazität führt zu einer höheren Gesamtauslastung ohne Geschäftsauswirkungen während des Normalbetriebs und hat den Nebeneffekt, dass die Kosten gesenkt werden.

# Bewährte Methoden in Bezug auf Nachhaltigkeit in der Cloud

Optimieren Sie die Workload-Platzierung und optimieren Sie Ihre Architektur im Hinblick auf Bedarf, Software, Daten, Hardware und Prozesse, um die Energieeffizienz zu erhöhen. Jeder dieser Bereiche bietet Möglichkeiten für den Einsatz bewährter Methoden zur Reduzierung der Nachhaltigkeitsauswirkungen Ihrer Cloud-Workloads – beispielsweise durch die Maximierung der Nutzung und die Minimierung von Abfällen sowie des Einsatzes der für Ihre Workloads bereitgestellten Energie und der verbrauchenden Ressourcen.

## Themen

- [Auswahl von Regionen](#)
- [Ausrichtung am Bedarf](#)
- [Software und Architektur](#)
- [Datenverwaltung](#)
- [Hardware und Services](#)
- [Prozess und Kultur](#)

## Auswahl von Regionen

Welche Region Sie für Ihren Workload auswählen, hat signifikante Auswirkungen auf seine KPIs, u. a. Leistung, Kosten und CO2-Bilanz. Um diese KPIs effizient zu verbessern, sollten Sie die Regionen für Ihren Workload abhängig von den Unternehmensanforderungen und Nachhaltigkeitszielen auswählen.

## Bewährte Methoden

- [SUS01-BP01 Auswählen der Region auf Grundlage von Unternehmensanforderungen und Nachhaltigkeitszielen](#)

## SUS01-BP01 Auswählen der Region auf Grundlage von Unternehmensanforderungen und Nachhaltigkeitszielen

Wählen Sie eine Region für Ihren Workload auf Grundlage Ihrer Geschäftsanforderungen und Nachhaltigkeitsvorgaben aus, um so KPIs wie Leistung, Kosten und CO2-Bilanz zu optimieren.

Typische Anti-Muster:

- Sie wählen die Region des Workloads auf Grundlage Ihres eigenen Standorts aus.
- Sie konsolidieren alle Workload-Ressourcen an einem geografischen Standort.

Vorteile der Einführung dieser bewährten Methode: Wenn Sie einen Workload in der Nähe von Amazon-Projekten für erneuerbare Energien oder in Regionen mit nachweislich niedrigen Kohlendioxidemissionen platzieren, kann die CO<sub>2</sub>-Bilanz eines Clouds-Workloads gesenkt werden.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: mittel

## Implementierungsleitfaden

Die AWS Cloud ist ein ständig wachsendes Netzwerk aus Regionen und Points of Presence (PoP), die durch eine globale Netzwerkinfrastruktur verbunden werden. Welche Region Sie für Ihren Workload auswählen, hat signifikante Auswirkungen auf seine KPIs, u. a. Leistung, Kosten und CO<sub>2</sub>-Bilanz. Um diese KPIs effizient zu verbessern, sollten Sie die Regionen für Ihren Workload abhängig von den Unternehmensanforderungen sowie Nachhaltigkeitszielen auswählen.

### Implementierungsschritte

- Befolgen Sie diese Schritte, um potenzielle Regionen für Ihren Workload zu bewerten und in die engere Auswahl zu nehmen. Berücksichtigen Sie dabei die Anforderungen Ihres Unternehmens, u. a. im Bezug auf Compliance, verfügbare Funktionen, Kosten und Latenz:
  - Vergewissern Sie sich, dass die Regionen konform sind und die entsprechenden lokalen Vorschriften erfüllen.
  - Prüfen Sie anhand der [Liste regionaler AWS-Services](#), ob die Regionen über die für Ihren Workload erforderlichen Services und Features verfügen.
  - Berechnen Sie die Kosten des Workloads in jeder Region mithilfe des [AWS Pricing Calculator](#).
  - Testen Sie die Netzwerklatenz zwischen den Standorten Ihrer Endbenutzer und jeder AWS-Region.
- Wählen Sie Regionen in der Nähe von Amazon-Projekten für erneuerbare Energien aus. Es sollte sich um Regionen handeln, in denen das Stromnetz nachweislich geringere Kohlendioxidemissionen generiert als andere Standorte (oder Regionen).
  - Ermitteln Sie die relevanten Nachhaltigkeitsrichtlinien, um die jährlichen CO<sub>2</sub>-Emissionen gemäß dem [Greenhouse Gas Protocol](#) zu nachzuverfolgen und zu vergleichen (marktbasierte und standortbasierte Verfahren).
  - Wählen Sie die Region entsprechend dem Verfahren aus, mit dem Sie CO<sub>2</sub>-Emissionen nachverfolgen. Weitere Informationen zum Auswählen einer Region anhand von

Nachhaltigkeitsrichtlinien finden Sie im [Artikel zum Auswählen einer Region für Ihren Workload auf Grundlage von Nachhaltigkeitszielen](#).

## Ressourcen

Zugehörige Dokumente:

- [Understanding your carbon emission estimations](#) (Grundlagen zu CO2-Emissionsschätzungen)
- [Amazon Weltweit](#)
- [Methodik für erneuerbare Energien](#)
- [„Relevante Aspekte bei der Wahl einer Region für Ihre Workloads“ erläutert](#)

Zugehörige Videos:

- [Nachhaltige Architektur und Reduzieren der AWS-CO2-Bilanz](#)

## Ausrichtung am Bedarf

Die Art und Weise, wie Benutzer und Anwendungen Ihre Workloads und andere Ressourcen nutzen, kann Sie bei der Identifizierung von Verbesserungen unterstützen, um Nachhaltigkeitsziele zu erreichen. Skalieren Sie Ihre Infrastruktur so, dass Sie den Bedarf kontinuierlich anpassen können. Sorgen Sie zudem dafür, dass zur Unterstützung Ihrer Benutzer nicht mehr Ressourcen verwendet werden als unbedingt nötig. Richten Sie Service-Levels an den Kundenanforderungen aus. Positionieren Sie Ressourcen so, dass die Netzwerkkapazitäten, die für Benutzer und Anwendungen erforderlich sind, begrenzt werden. Entfernen Sie ungenutzte Komponenten. Stellen Sie Teammitgliedern Geräte zur Verfügung, die ihre Anforderungen bei geringstmöglichen Auswirkungen auf die Nachhaltigkeit erfüllen.

Bewährte Methoden

- [SUS02-BP01 Dynamisches Skalieren der Workload-Infrastruktur](#)
- [SUS02-BP02 Ausrichten von SLAs an Nachhaltigkeitszielen](#)
- [SUS02-BP03 Beenden der Erstellung und Wartung nicht verwendeter Komponenten](#)
- [SUS02-BP04 Optimieren der geografischen Platzierung von Workloads auf der Grundlage ihrer Netzwerkanforderungen](#)

- [SUS02-BP05 Optimieren von Ressourcen für Teammitglieder im Hinblick auf die ausgeführten Aktivitäten](#)
- [SUS02-BP06 Implementierung von Pufferung oder Drosselung, um die Bedarfskurve zu verflachen](#)

## SUS02-BP01 Dynamisches Skalieren der Workload-Infrastruktur

Nutzen Sie die Elastizität der Cloud und skalieren Sie Ihre Infrastruktur dynamisch, um das Angebot an Cloud-Ressourcen an die Nachfrage anzupassen und eine Überbereitstellung bei Ihren Workloads zu vermeiden.

Typische Anti-Muster:

- Sie skalieren Ihre Infrastruktur nicht mit der Benutzerlast.
- Sie skalieren Ihre Infrastruktur stets manuell.
- Sie belassen die erhöhte Kapazität nach dem Hochskalieren, anstatt wieder herunterzuskalieren.

Vorteile der Einführung dieser bewährten Methode: Das Konfigurieren und Testen der Workload-Elastizität trägt dazu bei, das Angebot an Cloud-Ressourcen effizient an die Nachfrage anzupassen und eine Überbereitstellung von Kapazitäten zu vermeiden. Sie können die Vorteile der Elastizität in der Cloud nutzen, um die Kapazität während und nach Nachfragespitzen automatisch zu skalieren und so sicherzustellen, dass Sie nur die richtige Anzahl von Ressourcen nutzen, die für die Erfüllung Ihrer Geschäftsanforderungen erforderlich ist.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: mittel

### Implementierungsleitfaden

Die Cloud bietet Ihnen die Flexibilität, Ressourcen dynamisch durch verschiedene Mechanismen zu erweitern oder zu reduzieren, um einem veränderten Bedarf gerecht zu werden. Eine optimale Abstimmung von Angebot und Nachfrage führt zu den geringsten Auswirkungen auf die Umgebung für einen bestimmten Workload.

Die Nachfrage kann fest oder variabel sein und erfordert Metriken und Automatisierung, um sicherzustellen, dass die Verwaltung nicht zur Last wird. Anwendungen können vertikal (nach oben oder unten) skaliert werden, indem die Instance-Größe geändert wird, horizontal (nach innen oder außen), indem die Anzahl der Instances geändert wird, oder eine Kombination aus beidem.

Sie können verschiedene Ansätze nutzen, um das Angebot an Ressourcen auf die Nachfrage abzustimmen.

- Zielverfolgungsansatz: Überwachen Sie Ihre Skalierungsmetriken und erhöhen oder verringern Sie die Kapazität automatisch Ihrem Bedarf entsprechend.
- Prädiktives Skalieren: Skalieren Sie entsprechend der erwarteten täglichen und wöchentlichen Entwicklungen.
- Zeitplanbasierter Ansatz: Legen Sie Ihren eigenen Skalierungsplan entsprechend den vorhersehbaren Auslastungsänderungen fest.
- Service-Skalierung: Wählen Sie Services (wie Serverless), die nativ planmäßig skalierbar sind oder das Auto-Scaling als Funktion bieten.

Identifizieren Sie Zeiträume mit geringer oder gar keiner Nutzung und skalieren Sie Ressourcen, um überschüssige Kapazitäten zu entfernen und die Effizienz zu verbessern.

## Implementierungsschritte

- Elastizität ermöglicht das Anpassen der verfügbaren Ressourcen an den Bedarf. Instances, Container und Funktionen bieten Mechanismen für Elastizität, entweder in Kombination mit Auto-Scaling oder als Funktion des Services. AWS bietet eine Reihe von Mechanismen für das Auto-Scaling, um sicherzustellen, dass Workloads in Zeiten geringer Benutzerlast schnell und einfach herunterskaliert werden können. Hier sind einige Beispiele für Auto-Scaling-Mechanismen:

Auto scaling mechanism	Where to use
<a href="#">Amazon EC2 Auto Scaling</a>	Verwenden Sie diesen Mechanismus, um zu überprüfen, ob Sie die richtige Anzahl an Amazon EC2-Instances zur Verfügung haben, um die Benutzerlast für Ihre Anwendung zu bewältigen.
<a href="#">Application Auto Scaling</a>	Verwenden Sie diesen Mechanismus, um die Ressourcen für einzelne AWS-Services über Amazon EC2 hinaus automatisch zu skalieren, z. B. Lambda-Funktionen oder Amazon Elastic Container Service (Amazon ECS)-Services.

## Auto scaling mechanism

## Where to use

[Kubernetes Cluster Autoscaler](#)

Verwenden Sie diesen Mechanismus, um Kubernetes-Cluster in AWS automatisch zu skalieren.

- Das Skalieren wird häufig im Zusammenhang mit Datenverarbeitungsservices wie Amazon EC2-Instances oder AWS Lambda-Funktionen genannt. Ziehen Sie die Konfiguration von nicht Daten verarbeitenden Services wie [Amazon DynamoDB](#)-Lese- und Schreibkapazitätseinheiten oder [Amazon Kinesis Data Streams](#)-Shards in Betracht, um die Nachfrage zu decken.
- Prüfen Sie, ob die Metriken zum Hoch- oder Herunterskalieren für die jeweilige Art des bereitgestellten Workloads überprüft werden. Wenn Sie eine Anwendung zur Video-Transkodierung bereitstellen, wird eine CPU-Auslastung von 100 % erwartet, weshalb dies nicht die Hauptmetrik sein sollte. Sie können bei Bedarf eine [benutzerdefinierte Metrik](#) (wie etwa die Speichernutzung) für Ihre Skalierungsrichtlinie verwenden. Beachten Sie zur Auswahl der geeigneten Metriken die folgenden Hinweise zu Amazon EC2:
  - Es sollte sich um eine gültige Nutzungsmetrik handeln, die beschreibt, wie stark eine Instance genutzt wird.
  - Der Metrikwert muss proportional zur Anzahl der Instances in der Auto Scaling-Gruppe steigen oder sinken.
- Verwenden Sie für Ihre Auto Scaling-Gruppe eine [dynamische Skalierung](#) anstelle einer [manuellen Skalierung](#). Wir empfehlen außerdem, dass Sie bei der dynamischen Skalierung [Richtlinien zur Zielverfolgung](#) verwenden.
- Stellen Sie sicher, dass Workload-Bereitstellungen sowohl Hoch- als auch Herunterskalierungsereignisse verarbeiten können. Erstellen Sie Testszenarien für Herunterskalierungsereignisse, um zu überprüfen, ob sich der Workload wie erwartet verhält und die Benutzererfahrung nicht beeinträchtigt (z. B. Verlust von Sticky Sessions). Sie können die [Aktivitätshistorie](#) verwenden, um eine Skalierungsaktivität für eine Auto Scaling-Gruppe zu überprüfen.
- Evaluieren Sie Ihren Workload auf vorhersagbare Muster und skalieren Sie proaktiv, wenn Sie vorhergesagte und geplante Änderungen der Nachfrage erwarten. Mit der prädiktiven Skalierung können Sie die Notwendigkeit einer Überbereitstellung von Kapazität vermeiden. Weitere Einzelheiten finden Sie unter [Prädiktive Skalierung mit Amazon EC2 Auto Scaling](#).

## Ressourcen

### Zugehörige Dokumente:

- [Erste Schritte mit Amazon EC2 Auto Scaling](#)
- [Prädiktive Skalierung für EC2, unterstützt von Machine Learning](#)
- [Analyse des Benutzerverhaltens mit Amazon OpenSearch Service, Amazon Data Firehose und Kibana](#)
- [Was ist Amazon CloudWatch?](#)
- [Überwachen der DB-Last mit Performance Insights auf Amazon RDS](#)
- [Vorstellung von nativer Unterstützung für die prädiktive Skalierung mit Amazon EC2 Auto Scaling](#)
- [Vorstellung von Karpenter – Open-Source-Kubernetes-Cluster-Autoscaler mit hoher Leistung](#)
- [Detaillierte Einblicke in Amazon ECS Cluster Auto Scaling](#)

### Zugehörige Videos:

- [Entwickeln einer kosten-, energie- und ressourceneffizienten Datenverarbeitungsumgebung](#)
- [Bessere, schnellere und kostengünstigere Datenverarbeitung: Kostenoptimierung bei Amazon EC2 \(CMP202-R1\)](#)

### Zugehörige Beispiele:

- [Lab: Beispiele für Amazon EC2 Auto Scaling-Gruppen](#)
- [Lab: Implementierung von Autoscaling mit Karpenter](#)

## SUS02-BP02 Ausrichten von SLAs an Nachhaltigkeitszielen

Überprüfen und optimieren Sie die Service Level Agreements (SLA) für Workloads auf der Grundlage Ihrer Nachhaltigkeitsziele, um die für die Unterstützung Ihres Workloads erforderlichen Ressourcen zu minimieren und gleichzeitig die Geschäftsanforderungen zu erfüllen.

### Typische Anti-Muster:

- Workload-SLAs sind unbekannt oder nicht eindeutig.
- Sie definieren Ihre SLA nur für Verfügbarkeit und Leistung.



- Sie verwenden die gleichen Designmuster (wie Multi-AZ-Architektur) für alle Ihre Workloads.

Vorteile der Einführung dieser bewährten Methode: Die Abstimmung von SLAs mit Nachhaltigkeitszielen führt zu einer optimalen Ressourcennutzung bei gleichzeitiger Erfüllung der Geschäftsanforderungen.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: niedrig

## Implementierungsleitfaden

SLAs definieren das von einem Cloud-Workload erwartete Serviceniveau, z. B. Antwortzeit, Verfügbarkeit und Datenaufbewahrung. Sie beeinflussen die Architektur, die Ressourcennutzung und die Umweltauswirkungen eines Cloud-Workloads. Prüfen Sie regelmäßig die SLAs und gehen Sie Kompromisse ein, indem Sie die Ressourcennutzung in akzeptabler Weise verringern, um Auswirkungen auf die Nachhaltigkeit zu reduzieren.

## Implementierungsschritte

- Definieren oder ändern Sie SLAs, die Ihre Nachhaltigkeitsziele unterstützen und gleichzeitig Ihre geschäftlichen Anforderungen erfüllen, nicht darüber hinaus gehen.
- Gehen Sie Kompromisse ein, indem Sie Service Level in akzeptabler Weise verringern, um Auswirkungen auf die Nachhaltigkeit zu reduzieren.
  - Nachhaltigkeit und Zuverlässigkeit: Workloads mit hoher Verfügbarkeit verbrauchen in der Regel mehr Ressourcen.
  - Nachhaltigkeit und Leistung: Der Einsatz von mehr Ressourcen zur Leistungssteigerung könnte die Umwelt stärker belasten.
  - Nachhaltigkeit und Sicherheit: Übermäßig sichere Workloads könnten die Umwelt stärker belasten..
- Nutzen Sie Designmuster wie [Microservices auf AWS](#), die geschäftskritische Funktionen priorisieren, und lassen Sie für nicht kritische Funktionen niedrigere Service Level zu (z. B. für Reaktions- und Wiederherstellungszeiten).

## Ressourcen

Zugehörige Dokumente:

- [AWS Service Level Agreements \(SLAs\)](#)

- [Bedeutung von Dienstleistungsvereinbarungen für SaaS-Anbieter](#)

Zugehörige Videos:

- [Delivering sustainable, high-performing architectures](#) (Bereitstellung nachhaltiger, leistungsstarker Architekturen)
- [Build a cost-, energy-, and resource-efficient compute environment](#) (Entwickeln einer kosten-, energie- und ressourceneffizienten Datenverarbeitungsumgebung)

## SUS02-BP03 Beenden der Erstellung und Wartung nicht verwendeter Komponenten

Nehmen Sie nicht verwendete Ressourcen in Ihrem Workload außer Betrieb, um die Anzahl der Cloud-Ressourcen zu verringern, die zur Unterstützung Ihres Bedarfs und zur Minimierung von Verschwendung erforderlich sind.

Typische Anti-Muster:

- Sie analysieren Ihre Anwendung nicht auf Ressourcen, die redundant sind oder nicht mehr benötigt werden.
- Sie entfernen keine redundanten oder nicht mehr benötigten Ressourcen.

Vorteile der Nutzung dieser bewährten Methode: Das Entfernen nicht genutzter Ressourcen setzt Kapazitäten frei und verbessert die allgemeine Effizienz des Workloads.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: niedrig

### Implementierungsleitfaden

Nicht verwendete Ressourcen verbrauchen Cloud-Kapazitäten wie Speicherplatz oder Rechenleistung. Wenn Sie solche Ressourcen identifizieren und eliminieren, können Sie diese Kapazitäten freisetzen, was zu einer effizienteren Cloud-Architektur führt. Analysieren Sie Anwendungsressourcen (wie vorab kompilierte Berichte, Datensätze, statische Bilder) sowie Zugriffsmuster für Komponenten, um Redundanzen, eine zu geringe Auslastung und mögliche Kandidaten für die Außerbetriebnahme zu identifizieren. Entfernen Sie diese redundanten Ressourcen, um die Ressourcenverschwendung in Ihrem Workload zu reduzieren.

### Implementierungsschritte

- Verwenden Sie Überwachungstools zur Identifizierung statischer Ressourcen, die nicht mehr benötigt werden.
- Prüfen Sie vor dem Entfernen einer Ressource die Auswirkungen dieser Maßnahme auf die Architektur.
- Entwickeln Sie einen Plan und entfernen Sie Komponenten, die nicht mehr benötigt werden.
- Konsolidieren Sie sich überschneidende generierte Komponenten, um eine redundante Verarbeitung zu entfernen.
- Aktualisieren Sie Ihre Anwendungen, damit diese nicht mehr benötigte Ressourcen nicht weiter produzieren und speichern.
- Weisen Sie Dritte an, die Erstellung und Speicherung von Komponenten einzustellen, die in Ihrem Auftrag verwaltet und nicht mehr benötigt werden.
- Weisen Sie Dritte an, in Ihrem Auftrag erstellte redundante Komponenten zu konsolidieren.
- Prüfen Sie Ihren Workload regelmäßig und entfernen Sie nicht genutzte Ressourcen.

## Ressourcen

### Zugehörige Dokumente:

- [Optimizing your AWS Infrastructure for Sustainability, Part II: Storage](#) (Optimieren Ihrer AWS-Infrastruktur für Nachhaltigkeit, Teil II: Speicher)
- [How do I terminate active resources that I no longer need on my AWS-Konto?](#) (Wie beende ich aktive Ressourcen, die ich in meinem AWS-Konto nicht mehr benötige?)

### Zugehörige Videos:

- [How do I check for and then remove active resources that I no longer need on my AWS-Konto?](#) (Wie prüfe und entferne ich aktive Ressourcen, die ich in meinem AWS-Konto nicht mehr benötige?)

## SUS02-BP04 Optimieren der geografischen Platzierung von Workloads auf der Grundlage ihrer Netzwerkanforderungen

Wählen Sie Cloud-Standorte und -Services für Ihren Workload, die die Entfernungen reduzieren, über die Netzwerkdatenverkehr übertragen werden muss, um die Zahl der Netzwerkressourcen zu verringern, die zur Unterstützung Ihres Workloads erforderlich sind.

## Typische Anti-Muster:

- Sie wählen die Region des Workloads auf der Grundlage Ihres eigenen Standorts aus.
- Sie konsolidieren alle Workload-Ressourcen an einem geografischen Standort.
- Der gesamte Datenverkehr fließt durch Ihre bestehenden Rechenzentren.

Vorteile der Nutzung dieser bewährten Methode: Die Platzierung von Workloads in der Nähe der Benutzer bietet die geringstmögliche Latenz und verringert gleichzeitig die Bewegung der Daten durch das Netzwerk und damit die Umweltauswirkungen.

Risikostufe bei fehlender Befolgung dieser Best Practice: Mittel

## Implementierungsleitfaden

Die AWS Cloud-Infrastruktur basiert auf Standortoptionen wie etwa Regionen, Availability Zones, Platzierungsgruppen und Edge-Standorten wie [AWS Outposts](#) und [AWS Local Zones](#). Diese Standortoptionen stellen die Konnektivität zwischen Anwendungskomponenten, Cloud-Services, Edge-Netzwerken und On-Premises-Rechenzentren sicher.

Analysieren Sie die Netzwerkzugriffsmuster in Ihrem Workload, um festzustellen, wie diese verwendet werden können, um die Entfernungen für den Netzwerkdatenverkehr zu reduzieren.

## Implementierungsschritte

- Analysieren Sie die Netzwerkzugriffsmuster in Ihrem Workload, um zu ermitteln, wie die Benutzer Ihre Anwendung verwenden.
  - Verwenden Sie Überwachungstools wie [Amazon CloudWatch](#) und [AWS CloudTrail](#), um Daten über die Netzwerkaktivitäten zu sammeln.
  - Analysen Sie die Daten, um das Netzwerkzugriffsmuster zu identifizieren.
- Wählen Sie die Regionen für Ihre Workload-Bereitstellung auf der Grundlage der folgenden zentralen Elemente aus:
  - Ihr Nachhaltigkeitsziel: wie unter [Auswahl von Regionen erläutert](#).
  - Standort Ihrer Daten: Für datenintensive Anwendungen (wie etwa Big Data oder Machine Learning) sollte der Anwendungscode so nahe wie möglich zu den Daten ausgeführt werden.
  - Standort Ihrer Benutzer: Wählen Sie für benutzerseitige Anwendungen eine Region (oder Regionen) in der Nähe der Benutzer des Workloads.

- Weitere Einschränkungen: Berücksichtigen Sie auch Einschränkungen wie die Kosten und Compliance, wie unter [„Relevante Aspekte bei der Wahl einer Region für Ihre Workloads“](#) erläutert.
- Verwenden Sie lokale Zwischenspeicherung oder [AWS-Caching-Lösungen](#) für häufig genutzte Assets zur Verbesserung der Leistung, zur Verringerung der Datenbewegung und zur Reduzierung der Umweltauswirkungen.

Service	Verwendung
<a href="#">Amazon CloudFront</a>	Verwenden Sie dies für die Zwischenspeicherung statischer Inhalte wie Bilder, Skripts und Videos sowie dynamischer Inhalte wie API-Antworten oder Webanwendungen.
<a href="#">Amazon ElastiCache</a>	Verwenden Sie dies für die Zwischenspeicherung von Inhalten für Webanwendungen.
<a href="#">DynamoDB Accelerator</a>	Verwenden Sie dies für die Add-in-Speicher-Beschleunigung für Ihre DynamoDB-Tabellen.

- Nutzen Sie Services, die Ihnen dabei helfen können, Code näher an den Nutzern Ihres Workloads auszuführen:

Service	Verwendung
<a href="#">Lambda@Edge</a>	Verwenden Sie dies für rechenintensive Anwendungen, die initiiert werden, wenn sich Objekte nicht im Zwischenspeicher befinden.
<a href="#">Amazon CloudFront-Funktionen</a>	Verwenden Sie diese für einfache Anwendungsfälle wie HTTP(s)-Anfragen oder Antwortmanipulationen, die von kurzlebigen Funktionen initiiert werden können.
<a href="#">AWS IoT Greengrass</a>	Verwenden Sie dies für die Ausführung lokaler Rechenoperationen, Messaging sowie die

Service	Verwendung
	Datenzwischenspeicherung für verbundene Geräte.

- Nutzen Sie Verbindungspooling, um die erneute Nutzung von Verbindungen zu ermöglichen und die Zahl der erforderlichen Ressourcen zu reduzieren.
- Verwenden Sie verteilte Datenspeicher, die nicht auf persistente Verbindungen und synchrone Updates angewiesen sind, um regionale Benutzergruppen zu unterstützen.
- Ersetzen Sie vorab bereitgestellte statische Netzwerkkapazität durch geteilte dynamische Kapazitäten und teilen Sie die Auswirkungen von Netzwerkkapazitäten auf die Nachhaltigkeit mit anderen Abonnenten.

## Ressourcen

### Zugehörige Dokumente:

- [Optimieren Ihrer AWS-Infrastruktur für Nachhaltigkeit, Teil III: Netzwerke](#)
- [Amazon ElastiCache-Dokumentation](#)
- [Was ist Amazon CloudFront?](#)
- [Wichtigste Amazon CloudFront-Funktionen](#)

### Zugehörige Videos:

- [Demystifying data transfer on AWS \(Das Geheimnis der Datenübertragung in AWS lüften\)](#)
- [Scaling network performance on next-gen Amazon EC2 instances \(Skalierung der Netzwerkleistung auf EC2-Instances der nächsten Generation\)](#)

### Zugehörige Beispiele:

- [Workshops zu AWS-Netzwerken](#)
- [Nachhaltige Architektur — Minimierung des Datenverkehrs zwischen Netzwerken](#)

## SUS02-BP05 Optimieren von Ressourcen für Teammitglieder im Hinblick auf die ausgeführten Aktivitäten

Optimieren Sie die Ressourcen, die Teammitgliedern zur Verfügung gestellt werden, um negative Auswirkungen auf die Nachhaltigkeit zu minimieren und gleichzeitig ihre Anforderungen zu erfüllen.

Typische Anti-Muster:

- Sie berücksichtigen nicht die Auswirkungen der von Ihren Teammitgliedern verwendeten Geräte auf die Gesamteffizienz Ihrer Cloud-Anwendung.
- Sie verwalten und aktualisieren die von Team-Mitgliedern verwendeten Ressourcen manuell.

Vorteile der Nutzung dieser bewährten Methode: Die Optimierung der Teammitglieder-Ressourcen verbessert die allgemeine Effizienz Cloud-fähiger Anwendungen.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: niedrig

### Implementierungsleitfaden

Verstehen Sie die Ressourcen, mit denen Ihre Teammitglieder Ihre Services nutzen, deren erwartete Lebensdauer sowie die finanziellen und nachhaltigkeitsbezogenen Auswirkungen. Implementieren Sie Strategien zur Optimierung dieser Ressourcen. Beispielsweise können Sie komplexe Vorgänge wie Rendering und Kompilierung auf intensiv genutzter und skalierbarer Infrastruktur anstatt auf weniger ausgelasteten Einzelbenutzersystemen mit hohem Energieverbrauch ausführen.

### Implementierungsschritte

- Stellen Sie Workstations und andere Geräte entsprechend ihrer Verwendung bereit.
- Verwenden Sie virtuelle Desktops und Anwendungs-Streaming, um Upgrade- und Geräteanforderungen zu begrenzen.
- Verschieben Sie prozessor- oder arbeitsspeicherintensive Aufgaben in die Cloud, um deren Elastizität zu nutzen.
- Evaluieren Sie die Auswirkungen von Prozessen und Systemen auf die Lebenszyklen von Geräten. Wählen Sie Lösungen aus, die den Bedarf für Geräteauswachs Vorgänge minimieren und gleichzeitig die geschäftlichen Anforderungen erfüllen.
- Implementieren Sie die Remote-Verwaltung für Geräte, um die Zahl der Geschäftsreisen zu reduzieren.

- [AWS Systems Manager Fleet Manager](#) ist eine vereinheitlichte UI-Umgebung, die Ihnen dabei hilft, Ihre auf AWS oder On-Premises ausgeführten Knoten aus der Ferne zu überwachen.

## Ressourcen

Zugehörige Dokumente:

- [Was ist Amazon WorkSpaces?](#)
- [Kostenoptimierer für Amazon WorkSpaces](#)
- [Amazon AppStream 2.0 Documentation](#) (Dokumentation zu Amazon AppStream 2.0)
- [NICE DCV](#)

Zugehörige Videos:

- [Managing cost for Amazon WorkSpaces on AWS](#) (Verwalten der Kosten für Amazon WorkSpaces in AWS)

## SUS02-BP06 Implementierung von Pufferung oder Drosselung, um die Bedarfskurve zu verflachen

Pufferung und Drosselung verflachen die Bedarfskurve und reduzieren die erforderliche bereitgestellte Kapazität für Ihr Workload.

Typische Anti-Muster:

- Sie verarbeiten die Client-Anfragen sofort, obwohl dies nicht erforderlich ist.
- Sie analysieren die Anforderungen für Client-Anfragen nicht.

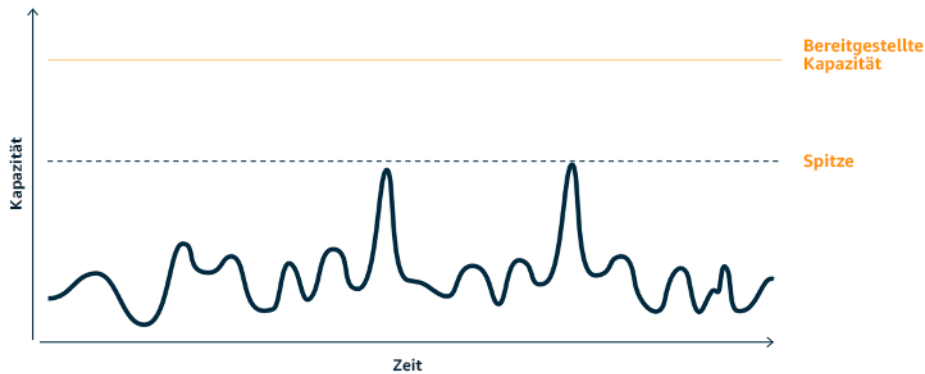
Vorteile der Nutzung dieser bewährten Methode: Das Verflachen der Bedarfskurve reduziert die erforderliche bereitgestellte Kapazität für den Workload. Die Reduzierung der bereitgestellten Kapazität bedeutet geringeren Energieverbrauch und geringere Umweltauswirkungen.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: niedrig

Die Verflachung der Bedarfskurve kann Ihnen dabei helfen, die bereitgestellte Kapazität für einen Workload zu verringern und dessen Umweltauswirkungen zu reduzieren. Nehmen wir einen Workload

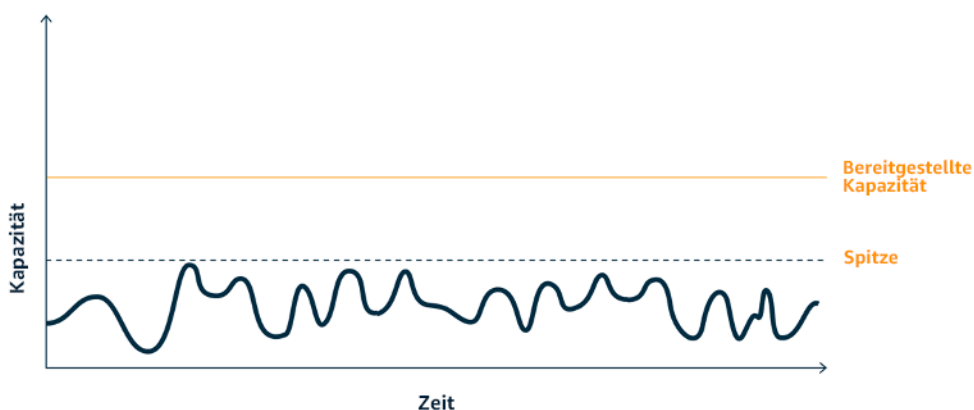


mit der nachfolgend gezeigten Bedarfskurve. Dieser Workload hat zwei Spitzen und um damit umzugehen, wird die Ressourcenkapazität bereitgestellt, die hier durch die orangefarbene Linie angezeigt wird. Die für diesen Workload aufgewendeten Ressourcen und die eingesetzte Energie werden nicht durch die Fläche unter der Bedarfskurve, sondern von der Linie für die bereitgestellte Kapazität angezeigt, da für den Umgang mit den beiden Spitzen bereitgestellte Kapazität erforderlich ist.



Bedarfskurve mit zwei deutlichen Spitzen, die hohe bereitgestellte Kapazität erfordern.

Sie können Pufferung oder Drosselung verwenden, um die Bedarfskurve zu beeinflussen und die Spitzen abzumildern, was weniger bereitgestellte Kapazität und einen geringeren Energieverbrauch bedeutet. Implementieren Sie Drosselung, wenn Ihre Clients wiederholte Versuche durchführen können. Implementieren Sie Pufferung, um die Anforderung zu speichern und die Verarbeitung auf einen späteren Zeitpunkt zu verschieben.



Auswirkungen des Drosselns auf die Bedarfskurve und die bereitgestellte Kapazität.

## Implementierungsschritte

- Analysieren Sie die Client-Anfragen, um festzulegen, wie darauf zu reagieren ist. Wichtige Faktoren dabei sind:
  - Kann diese Anfrage in asynchroner Weise verarbeitet werden?
  - Kann der Client die Anfrage erneut versuchen?
- Wenn dies der Fall ist, können Sie Drosselung verwenden, die der Quelle mitteilt, dass wenn sie die Anfrage zum aktuellen Zeitpunkt nicht bedienen kann, es später erneut versucht werden sollte.
  - Sie können [Amazon API Gateway](#) verwenden, um Drosselung zu implementieren.
- Für Clients, die Anfragen nicht erneut versuchen können, muss zur Verflachung der Bedarfskurve ein Puffer implementiert werden. Ein Puffer verschiebt die Anforderungsverarbeitung, so dass Anwendungen, die mit unterschiedlichen Raten ausgeführt werden, effektiv kommunizieren können. Bei der Pufferung werden Nachrichten von Produzenten in eine Warteschlange oder einen Stream gestellt. Nachrichten können dadurch von Verbrauchern in der für ihre Geschäftsanforderungen passenden Geschwindigkeit gelesen und verarbeitet werden.
  - [Amazon Simple Queue Service \(Amazon SQS\)](#) ist ein verwalteter Service, der Warteschlangen bietet, die es einem einzelnen Verbraucher ermöglichen, individuelle Nachrichten zu lesen.
  - [Amazon Kinesis](#) stellt einen Stream bereit, der es vielen Verbrauchern ermöglicht, dieselben Nachrichten zu lesen.
- Analysieren Sie den Gesamtbedarf, die Änderungsrate und die erforderliche Reaktionszeit, um die korrekte Größe der erforderlichen Drosselung oder des Puffers zu bestimmen.

## Ressourcen

### Zugehörige Dokumente:

- [Erste Schritte mit Amazon SQS](#)
- [Application integration Using Queues and Messages](#) (Anwendungsintegration mit Warteschlangen und Nachrichten)

### Zugehörige Videos:

- [Choosing the Right Messaging Service for Your Distributed App](#) (Den richtigen Messaging-Service für Ihre verteilte App auswählen)

# Software und Architektur

Implementieren Sie Muster für den Lastausgleich und die Wahrung einer konsistent hohen Nutzung der bereitgestellten Ressourcen, um die Zahl der genutzten Ressourcen zu minimieren. Komponenten werden möglicherweise aufgrund von Änderungen des Benutzerverhaltens über die Zeit nicht mehr genutzt. Prüfen Sie Muster und Architekturen, um nicht ausreichend genutzte Komponenten zu konsolidieren und so die Nutzung insgesamt zu erhöhen. Nehmen Sie Komponenten außer Betrieb, die nicht mehr benötigt werden. Identifizieren Sie die Leistung Ihrer Workload-Komponenten und optimieren Sie die Komponenten, die die meisten Ressourcen verbrauchen. Achten Sie auf die Geräte, mit denen Ihre Kunden auf Ihre Services zugreifen, und implementieren Sie Muster, um den Bedarf für Geräte-Upgrades zu minimieren.

## Bewährte Methoden

- [SUS03-BP01 Optimieren von Software und Architektur für asynchrone und geplante Aufträge](#)
- [SUS03-BP02 Entfernen oder Refaktorisieren von Workload-Komponenten mit geringer oder keiner Nutzung](#)
- [SUS03-BP03 Optimieren von Codebereichen, die die meiste Zeit oder die meisten Ressourcen verbrauchen](#)
- [SUS03-BP04 Optimieren der Auswirkungen auf Geräte und Ausrüstung von Kunden](#)
- [SUS03-BP05 Verwenden von Softwaremustern und Architekturen, die Datenzugriffs- und Speichermuster optimal unterstützen](#)

## SUS03-BP01 Optimieren von Software und Architektur für asynchrone und geplante Aufträge

Verwenden Sie effiziente Software- und Architekturmuster wie warteschlangenbasierte Systeme, um eine durchgängig hohe Auslastung von bereitgestellten Ressourcen zu erzielen.

### Typische Anti-Muster:

- Sie stellen zu viele Ressourcen im Cloud-Workload bereit, um auf unerwartete Nachfragesteigerungen reagieren zu können.
- In Ihrer Architektur werden Absender und Empfänger von asynchronen Nachrichten nicht durch eine Messaging-Komponente entkoppelt.

### Vorteile der Nutzung dieser bewährten Methode:

- Durch effiziente Software- und Architekturmuster werden ungenutzte Ressourcen in Ihrem Workload minimiert und die allgemeine Effizienz gesteigert.
- Sie können die Verarbeitung unabhängig vom Empfang asynchroner Nachrichten skalieren.
- Durch eine Messaging-Komponente gelten weniger strenge Verfügbarkeitsanforderungen, die mit weniger Ressourcen erfüllt werden können.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: Mittel

## Implementierungsleitfaden

Verwenden Sie effiziente Architekturmuster wie eine [ereignisgesteuerte Architektur](#), die zu einer gleichmäßigen Nutzung der Komponenten führen und die Überbereitstellung in Ihrem Workload minimieren. Durch die Verwendung effizienter Architekturmuster werden ungenutzte Ressourcen, die aufgrund von Änderungen der Nachfrage im Laufe der Zeit nicht genutzt werden, minimiert.

Analysieren Sie die Anforderungen Ihrer Workload-Komponenten und führen Sie Architekturmuster ein, mit denen die allgemeine Auslastung der Ressourcen gesteigert wird. Nehmen Sie Komponenten außer Betrieb, die nicht mehr benötigt werden.

## Implementierungsschritte

- Analysieren Sie die Nachfrage für Ihren Workload, um zu bestimmen, wie diese erfüllt werden kann.
- Verwenden Sie für Anfragen oder Aufträge, für die keine synchronen Antworten erforderlich sind, warteschlangenbasierte Architekturen und Worker mit automatischer Skalierung, durch die die Auslastung maximiert wird. Hier finden Sie einige Beispiele für Situationen, in denen Sie eine warteschlangenbasierte Architektur in Erwägung ziehen sollten:

Queuing mechanism	Description
<a href="#">AWS Batch-Warteschlangen</a>	AWS Batch-Aufträge werden an eine Auftragswarteschlange gesendet, in der sie bleiben, bis ihre Ausführung in einer Datenverarbeitungsumgebung geplant werden kann.

Queuing mechanism	Description
<a href="#">Amazon Simple Queue Service und Amazon EC2 Spot Instances</a>	Durch das Koppeln von Amazon SQS und Spot Instances lassen sich fehlertolerante und effiziente Architekturen erstellen.

- Verwenden Sie für Anfragen oder Aufträge, die jederzeit verarbeitet werden können, Planungsmechanismen zur Auftragsverarbeitung in Batches, um die Effizienz zu steigern. Hier sind einige Beispiele für Planungsmechanismen in AWS:

Scheduling mechanism	Description
<a href="#">Amazon EventBridge Scheduler</a>	Eine <a href="#">Amazon EventBridge</a> -Funktion, mit der Sie in großem Umfang geplante Aufgaben erstellen, ausführen und verwalten können.
<a href="#">Zeitbasierte AWS Glue-Pläne</a>	Hiermit definieren Sie einen zeitbasierten Plan für Crawler und Aufträge in AWS Glue.
<a href="#">Geplante Amazon Elastic Container Service (Amazon ECS)-Aufgaben</a>	Amazon ECS unterstützt das Erstellen von geplanten Aufgaben. Bei geplanten Aufgaben werden mit Amazon EventBridge-Regeln Aufgaben nach einem Zeitplan oder als Reaktion auf ein EventBridge-Ereignis ausgeführt.
<a href="#">Instance Scheduler</a>	Konfigurieren Sie Zeitpläne zum Starten und Beenden Ihrer Amazon EC2- und Amazon Relational Database Service-Instances.

- Wenn Sie Abfrage- und Webhook-Mechanismen in Ihrer Architektur verwenden, ersetzen Sie diese durch Ereignisse. Erstellen Sie mit [ereignisgesteuerten Architekturen](#) hocheffiziente Workloads.
- Nutzen Sie [Serverless on AWS](#), um eine übermäßige Bereitstellung in einer Infrastruktur zu eliminieren.
- Wählen Sie die richtige Größe für Ihre Architektur, um zu vermeiden, dass ungenutzte Ressourcen auf Eingaben warten.

## Ressourcen

Zugehörige Dokumente:

- [Was ist Amazon Simple Queue Service?](#)
- [Was ist Amazon MQ?](#)
- [Scaling based on Amazon SQS](#) (Skalierung auf Grundlage von Amazon SQS)
- [Was ist AWS Step Functions?](#)
- [Was ist AWS Lambda?](#)
- [Using AWS Lambda with Amazon SQS](#) (Verwenden von Lambda mit Amazon SQS)
- [Was ist Amazon EventBridge?](#)

Zugehörige Videos:

- [Moving to event-driven architectures](#) (Umstieg auf ereignisgesteuerte Architekturen)

## SUS03-BP02 Entfernen oder Refaktorisieren von Workload-Komponenten mit geringer oder keiner Nutzung

Entfernen Sie ungenutzte Komponenten, die nicht mehr benötigt werden, und refaktorisieren Sie Komponenten mit geringer Nutzung, um die Verschwendung von Ressourcen zu begrenzen.

Typische Anti-Muster:

- Sie prüfen den Nutzungsgrad der einzelnen Komponenten Ihres Workloads nicht regelmäßig.
- Sie prüfen und analysieren nicht die Empfehlungen von AWS-Dimensionierungstools wie etwa [AWS Compute Optimizer](#).

Vorteile der Nutzung dieser bewährten Methode: Das Entfernen nicht genutzter Komponenten minimiert Ausschuss und verbessert die allgemeine Effizienz Ihres Workloads.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: Mittel

### Implementierungsleitfaden

Prüfen Sie Ihren Workload, um nicht oder wenig genutzte Komponenten zu identifizieren. Dies ist ein sich wiederholender Verbesserungsprozess, der von Änderungen beim Bedarf oder

der Einführung eines neuen Cloud-Services ausgelöst werden kann. Beispielsweise kann ein deutliches Zurückgehen der Ausführungszeit der [AWS Lambda](#)-Funktion darauf hindeuten, dass die Speichergröße reduziert werden muss. Oder wenn AWS neue Services und Funktionen veröffentlicht, können sich die optimalen Services und die Architektur für Ihren Workload ändern.

Überwachen Sie kontinuierlich die Workload-Aktivität und suchen Sie nach Möglichkeiten zur Verbesserung des Nutzungsgrads einzelner Komponenten. Wenn Sie nicht genutzte Komponenten entfernen und Dimensionierungsaktivitäten durchführen, erreichen Sie Ihre geschäftlichen Ziele mit der geringstmöglichen Menge von Cloud-Ressourcen.

### Implementierungsschritte

- Überwachen und erfassen Sie die Nutzungsmetriken für kritische Komponenten Ihres Workloads (etwa CPU-Nutzung, Speichernutzung oder Netzwerkdurchsatz in [Amazon CloudWatch-Metriken](#)).
- Prüfen Sie für stabile Workloads regelmäßig AWS-Dimensionierungstools wie [AWS Compute Optimizer](#), um nicht oder wenig genutzte Komponenten zu identifizieren.
- Prüfen Sie für kurzzeitige Workloads die Nutzungsmetriken, um nicht oder wenig genutzte Komponenten zu identifizieren.
- Nehmen Sie nicht mehr benötigte und dazugehörige Ressourcen (wie etwa Amazon ECR-Images) außer Betrieb.
- Konsolidieren oder refaktorisieren Sie nicht ausreichend genutzte Ressourcen mit anderen Ressourcen, um die Nutzungseffizienz zu verbessern. Sie können beispielsweise mehrere kleine Datenbanken auf einer einzelnen [Amazon RDS](#)-Datenbank-Instance bereitstellen, anstatt Datenbanken auf einzelnen sehr wenig ausgenutzten Instances auszuführen.
- Verstehen Sie die [Ressourcen, die Ihr Workload für die Durchführung einer Arbeitseinheit bereitstellt](#).

### Ressourcen

Zugehörige Dokumente:

- [AWS Trusted Advisor](#)
- [Was ist Amazon CloudWatch?](#)
- [Automated Cleanup of Unused Images in Amazon ECR](#) (Automatische Bereinigung von nicht verwendeten Images in Amazon ECR)

Zugehörige Beispiele:

- [Well-Architected Lab – Dimensionierung mit AWS Compute Optimizer](#)
- [Well-Architected Lab – Optimieren von Hardwaremustern und Überwachen von KPIs zur Nachhaltigkeit](#)

## SUS03-BP03 Optimieren von Codebereichen, die die meiste Zeit oder die meisten Ressourcen verbrauchen

Optimieren Sie den Code, der innerhalb der verschiedenen Komponenten Ihrer Architektur ausgeführt wird, um die Ressourcennutzung zu minimieren und die Leistung zu maximieren.

Typische Anti-Muster:

- Sie versäumen die Optimierung Ihres Codes für die Ressourcennutzung.
- Sie reagieren auf Leistungsprobleme normalerweise mit Erhöhung des Ressourceneinsatzes.
- Ihr Code-Prüfungs- und -Entwicklungsprozess verfolgt keine Leistungsänderungen.

Vorteile der Nutzung dieser bewährten Methode: Effizienter Code minimiert den Ressourcenverbrauch und verbessert die Leistung.

Risikostufe bei fehlender Befolgung dieser Best Practice: Mittel

### Implementierungsleitfaden

Es ist sehr wichtig, jeden funktionalen Bereich, einschließlich des Codes einer für die Cloud erstellten Anwendung, zu untersuchen, um ihre Ressourcennutzung und Leistung zu optimieren. Überwachen Sie kontinuierlich die Leistung Ihres Workloads in Build-Umgebungen und Produktionsbereichen und suchen Sie nach Möglichkeiten, Code-Snippets zu verbessern, die einen besonders hohen Ressourcenverbrauch haben. Führen Sie einen regelmäßigen Prüfungsprozess ein, um Fehler oder Anti-Muster in Ihrem Code zu identifizieren, die Ressourcen in ineffizienter Weise nutzen. Nutzen Sie einfache und effiziente Algorithmen, die dieselben Ergebnisse für Ihre Anwendungsfälle liefern.

### Implementierungsschritte

- Führen Sie bei der Entwicklung Ihrer Workloads einen automatischen Code-Prüfungsprozess ein, um die Qualität zu verbessern sowie Fehler und Anti-Muster zu identifizieren.



- [Automatisieren von Codeüberprüfungen mit Amazon CodeGuru Reviewer](#)
- [Erkennen von Concurrency-Fehlern mit Amazon CodeGuru](#)
- [Verbessern der Codequalität für Python-Anwendungen mit Amazon CodeGuru](#)
- Überwachen Sie bei der Ausführung Ihrer Workloads die Ressourcen, um Komponenten mit einem hohen Ressourcenbedarf pro Arbeitseinheit als Ziele für Code-Prüfungen zu identifizieren.
- Verwenden Sie einen Code-Profiler für Code-Prüfungen, um die Codebereiche als Optimierungsziele zu identifizieren, die die meiste Zeit oder die meisten Ressourcen verwenden.
  - [Reduzieren des CO2-Fußabdrucks Ihrer Organisation mit Amazon CodeGuru Profiler](#)
  - [Verständnis der Speichernutzung in Ihrer Java-Anwendung mit Amazon CodeGuru Profiler](#)
  - [Verbessern des Kundenkomforts und Senken von Kosten mit Amazon CodeGuru Profiler](#)
- Verwenden Sie das jeweils effizienteste Betriebssystem und die optimale Programmiersprache für den Workload. Weitere Informationen zu energieeffizienten Programmiersprachen (einschließlich Rust) finden Sie unter [Nachhaltigkeit mit Rust](#).
- Ersetzen Sie rechenintensive Algorithmen durch einfachere und effizientere Versionen, die dieselben Ergebnisse liefern.
- Entfernen Sie unnötigen Code und überflüssige Formatierungen.

## Ressourcen

### Zugehörige Dokumente:

- [Was ist Amazon CodeGuru Profiler?](#)
- [FPGA-Instances](#)
- [Die AWS SDKs für die Entwicklung in AWS](#)

### Zugehörige Videos:

- [Improve Code Efficiency Using Amazon CodeGuru Profiler \(Verbessern der Code-Effizienz mit Amazon CodeGuru Profiler\)](#)
- [Automate Code Reviews and Application Performance Recommendations with Amazon CodeGuru \(Automatisieren von Codeprüfungen und Empfehlungen zur Anwendungsleistung mit Amazon CodeGuru\)](#)

## SUS03-BP04 Optimieren der Auswirkungen auf Geräte und Ausrüstung von Kunden

Verstehen Sie die in Ihrer Architektur verwendeten Geräte und nutzen Sie Strategien, um ihre Nutzung zu reduzieren. Dies kann die Umweltauswirkungen Ihres Cloud-Workloads insgesamt verringern.

Typische Anti-Muster:

- Sie ignorieren die Umweltauswirkungen der Geräte, die Ihre Kunden verwenden.
- Sie verwalten und aktualisieren die von Kunden verwendeten Ressourcen manuell.

Vorteile der Nutzung dieser bewährten Methode: Die Implementierung von Softwaremustern und Funktionen, die für Kundengeräte optimiert sind, können die Umweltauswirkungen des Cloud-Workloads insgesamt verringern.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: Mittel

### Implementierungsleitfaden

Die Implementierung für Kundengeräte optimierter Softwaremuster und Funktionen können die Umweltauswirkungen auf unterschiedliche Weise reduzieren:

- Die Implementierung neuer abwärtskompatibler Funktionen kann die Anzahl der Hardwareaustauschvorgänge verringern.
- Die Optimierung einer Anwendung, so dass sie effizient auf Geräten ausgeführt werden kann, kann bei der Reduzierung des Energieverbrauchs helfen und die Batterielaufzeit verlängern (falls Batterien zum Einsatz kommen).
- Die Optimierung einer Anwendung für Geräte kann auch Datenübertragungen über das Netzwerk verringern.

Verstehen Sie die in Ihrer Architektur verwendeten Geräte, ihre erwartete Lebensdauer und die Auswirkungen des Austauschs dieser Komponenten. Implementieren Sie Softwaremuster und Funktionen, die dabei helfen, den Energieverbrauch von Geräten zu senken, und den Austausch von Geräten sowie manuelle Upgrades durch Kunden seltener erforderlich machen.

### Implementierungsschritte

- Inventarisieren Sie die in ihrer Architektur verwendeten Geräte. Dabei kann es sich um Mobilgeräte, Tablets, IOT-Geräte, Smart Light- oder auch Smartgeräte in einer Fabrik handeln.
- Optimieren Sie die auf den Geräten ausgeführte Anwendung:
  - Verwenden Sie Strategien wie die Ausführung von Aufgaben im Hintergrund, um den Energieverbrauch zu verringern.
  - Berücksichtigen Sie beim Erstellen von Nutzlasten Netzwerkbandbreite und Latenz und implementieren Sie Funktionen, mit denen Ihre Anwendungen auch über Verbindungen mit geringer Bandbreite und hoher Latenz gut funktionieren.
  - Wandeln Sie Payloads und Dateien in von den Geräten benötigte optimierte Formate um. Sie können beispielsweise [Amazon Elastic Transcoder](#) oder [AWS Elemental MediaConvert](#) verwenden, um große, qualitativ hochwertige Digitalmediendateien in Formate umzuwandeln, die Benutzer auf Mobilgeräten abspielen können.
  - Führen Sie rechenintensive Aktivitäten (z. B. das Rendern von Bildern) serverseitig aus oder nutzen Sie Anwendungs-Streaming, um den Benutzerkomfort auf älteren Geräten zu verbessern.
  - Segmentieren und paginieren Sie Ausgaben, besonders für interaktive Sitzungen, um Nutzlasten zu verwalten und lokale Speicheranforderungen zu begrenzen.
- Verwenden Sie einen automatisierten Over-the-Air (OTA)-Mechanismus, um Aktualisierungen für ein oder mehrere Geräte bereitzustellen.
  - Mit einer [CI/CD-Pipeline](#) können Sie mobile Anwendungen aktualisieren.
  - Mit [AWS IoT Device Management](#) können Sie verbundene Geräte in großem Umfang aus der Ferne verwalten.
- Verwenden Sie zum Testen neuer Funktionen und Aktualisierungen verwaltete Gerätefarmen mit repräsentativen Sätzen von Hardwaregeräten, um den Umfang der unterstützten Geräte zu maximieren. Weitere Informationen finden Sie in [SUS06-BP04 Verwenden verwalteter Gerätefarmen für Tests](#).

## Ressourcen

### Zugehörige Dokumente:

- [What is AWS Device Farm?](#) (Was ist AWS Device Farm?)
- [Amazon AppStream 2.0 Documentation](#) (Dokumentation zu Amazon AppStream 2.0)
- [NICE DCV](#)
- [OTA-Tutorial zur Aktualisierung der Firmware auf Geräten mit FreeRTOS](#)

Zugehörige Videos:

- [Introduction to AWS Device Farm](#)(Einführung in AWS Device Farm)

## SUS03-BP05 Verwenden von Softwaremustern und Architekturen, die Datenzugriffs- und Speichermuster optimal unterstützen

Identifizieren Sie, wie Daten in Ihrem Workload verwendet, von Benutzern genutzt, übertragen und gespeichert werden. Verwenden Sie Softwaremuster und Architekturen, die den Datenzugriff und die Speicherung optimal unterstützen, um die zur Unterstützung des Workloads erforderlichen Computing-, Netzwerk- und Speicherressourcen zu reduzieren.

Typische Anti-Muster:

- Sie gehen davon aus, dass für alle Workloads ähnliche Datenspeicher- und Zugriffsmuster gelten.
- Sie verwenden nur eine Speicherebene, vorausgesetzt, dass alle Workloads in diese Ebene passen.
- Sie gehen davon aus, dass Datenzugriffsmuster im Laufe der Zeit konsistent bleiben.
- Ihre Architektur unterstützt potenzielle hohe Bursts beim Datenzugriff, was dazu führt, dass die Ressourcen die meiste Zeit ungenutzt bleiben.

Vorteile der Nutzung dieser bewährten Methode: Die Auswahl und Optimierung Ihrer Architektur auf der Grundlage von Datenzugriffs- und Speichermustern hilft bei der Reduzierung der Entwicklungskomplexität und der Steigerung der allgemeinen Nutzung. Das Verständnis, wann globale Tabellen, Datenpartitionen und Caching verwendet werden sollen, hilft Ihnen dabei, den Betriebsaufwand zu verringern und basierend auf Ihren Workload-Anforderungen zu skalieren.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: Mittel

### Implementierungsleitfaden

Verwenden Sie Software- und Architekturmuster, die optimal zu den Eigenschaften Ihrer Daten und den Zugriffsmustern passen. Verwenden Sie etwa eine [moderne Datenarchitektur auf AWS](#), die die Nutzung speziell erstellter Services ermöglicht, die für Ihre ganz speziellen Analyseanwendungsfälle optimiert sind. Diese Architekturmuster ermöglichen die effiziente Datenverarbeitung und verringern die Ressourcennutzung.

### Implementierungsschritte

- Analysieren Sie die Eigenschaften ihrer Daten und Ihre Zugriffsmuster, um die korrekte Konfiguration für Ihre Cloud-Ressourcen zu identifizieren. Zu den berücksichtigenden Schlüsselmerkmalen gehören:
  - Datentyp: strukturiert, semistrukturiert, unstrukturiert
  - Datenwachstum: begrenzt, unbegrenzt
  - Lebensdauer von Daten: anhaltend, flüchtig, vorübergehend
  - Zugriffsmuster: Lese- oder Schreibzugriff, Häufigkeit von Aktualisierungen, schwankend oder konsistent
- Verwenden Sie Architekturmuster, die Datenzugriffs- und Speichermuster optimal unterstützen.
  - [Let's Architect! Moderne Datenarchitekturen](#)
  - [Datenbanken auf AWS: Das richtige Tool für jede Aufgabe](#)
- Nutzen Sie Technologien, die nativ mit komprimierten Daten funktionieren.
- Verwenden Sie zweckgerichtet erstellte [Analyseservices](#) für die Datenverarbeitung in Ihrer Architektur.
- Verwenden Sie die Datenbank-Engine, die das dominierende Abfragemuster jeweils am besten unterstützt. Verwalten Sie Ihre Datenbankindizes so, dass sie die effiziente Ausführung von Abfragen unterstützen. Weitere Informationen finden Sie unter [AWS-Datenbanken](#).
- Wählen Sie Netzwerkprotokolle aus, die die Menge der genutzten Netzwerkkapazitäten in Ihrer Architektur reduzieren.

## Ressourcen

### Zugehörige Dokumente:

- [Athena Compression Support file formats](#) (Athena-Komprimierungs-Support-Dateiformate)
- [COPY aus spaltenbasierten Datenformaten mit Amazon Redshift](#)
- [Converting Your Input Record Format in Firehose](#) (Umwandeln Ihres Eingabedatensatzformats in Firehose)
- [Format Options for ETL Inputs and Outputs in AWS Glue](#) (Formatierungsoptionen für ETL-Eingaben und -Ausgaben in AWS Glue)
- [Improve query performance on Amazon Athena by Converting to Columnar Formats](#) (Verbessern der Abfrageleistung in Amazon Athena durch Umwandlung in Spaltenformate)
- [Laden komprimierter Datendateien aus Amazon S3 mit Amazon Redshift](#)

- [Überwachung der DB-Last mit Performance Insights auf Amazon Aurora](#)
- [Überwachung der DB-Last mit Performance Insights auf Amazon RDS](#)
- [Amazon S3 Intelligent-Tiering storage class](#) (Amazon S3-Intelligent-Tiering-Speicherklasse)

Zugehörige Videos:

- [Building modern data architectures on AWS](#) (Erstellen von modernen Datenarchitekturen in AWS)

## Datenverwaltung

Implementieren Sie Verfahren für die Datenverwaltung, die den zur Unterstützung Ihres Workloads bereitgestellten Speicher und die für dessen Nutzung erforderlichen Ressourcen reduzieren. Identifizieren Sie Ihre Daten und verwenden Sie Speichertechnologien und Konfigurationen, die den Unternehmenswert und die Nutzung der Daten optimal unterstützen. Verschieben Sie die Daten während des Lebenszyklus zu effizienteren Speichern mit geringerer Leistung, wenn die Anforderungen abnehmen. Löschen Sie Daten, die nicht mehr benötigt werden.

Bewährte Methoden

- [SUS04-BP01 Implementieren einer Richtlinie für die Klassifizierung von Daten](#)
- [SUS04-BP02 Verwenden von Technologien, die Datenzugriff und Speichermuster unterstützen](#)
- [SUS04-BP03 Verwalten des Lebenszyklus von Datensätzen mithilfe von Richtlinien](#)
- [SUS04-BP04 Verwendung von Elastizität und Automatisierung zur Erweiterung des Block-Speichers oder des Dateisystems](#)
- [SUS04-BP05 Entfernen nicht benötigter oder redundanter Daten](#)
- [SUS04-BP06 Verwenden geteilter Dateisysteme oder Objektspeicher für den Zugriff auf allgemeine Daten](#)
- [SUS04-BP07 Minimieren von Datenübertragungen zwischen Netzwerken](#)
- [SUS04-BP08 Sichern von Daten nur in dem Fall, wenn ihre erneute Erstellung schwierig ist](#)

### SUS04-BP01 Implementieren einer Richtlinie für die Klassifizierung von Daten

Klassifizieren Sie die Daten, um zu verstehen, wie wichtig sie für die Geschäftsergebnisse sind, und wählen Sie die richtige energieeffiziente Speicherebene zur Speicherung der Daten.

## Typische Anti-Muster:

- Sie identifizieren keine Datenbestände mit ähnlichen Merkmalen (z. B. Sensibilität, Geschäftskritikalität oder gesetzliche Anforderungen), die verarbeitet oder gespeichert werden.
- Sie haben keinen Datenkatalog zur Inventarisierung Ihrer Datenbestände eingeführt.

Vorteile der Nutzung dieser bewährten Methode: Durch die Implementierung einer Datenklassifizierungsrichtlinie können Sie die energieeffizienteste Speicherebene für Daten bestimmen.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: Mittel

## Implementierungsleitfaden

Bei der Datenklassifizierung wird identifiziert, welche Arten von Daten in einem Informationssystem verarbeitet und gespeichert werden, das einer Organisation gehört oder von ihr betrieben wird. Dazu gehört auch die Bestimmung der Kritikalität der Daten und der wahrscheinlichen Auswirkungen von Preisgaben, Verlusten oder Missbrauch von Daten.

Implementieren Sie Richtlinien zur Datenklassifizierung, indem Sie von der kontextuellen Verwendung der Daten ausgehen und ein Kategorisierungsschema erstellen, das den Grad der Kritikalität eines bestimmten Datensatzes für die Abläufe eines Unternehmens berücksichtigt.

## Implementierungsschritte

- Führen Sie eine Bestandsaufnahme der verschiedenen Datentypen durch, die für Ihren Workload vorhanden sind.
  - Einzelheiten zu den Kategorien für die Datenklassifizierung finden Sie im [Data Classification Whitepaper](#).
- Bestimmen Sie die Kritikalität, Vertraulichkeit, Integrität und Verfügbarkeit von Daten auf der Grundlage des Risikos für das Unternehmen. Verwenden Sie diese Anforderungen, um Daten in eine der von Ihnen gewählten Datenklassifizierungsebenen einzuteilen.
  - Ein Beispiel finden Sie unter [Four simple steps to classify your data and secure your startup](#) (Vier einfache Schritte zur Klassifizierung Ihrer Daten und zur Sicherung Ihres Startups).
- Prüfen Sie die Umgebung regelmäßig auf nicht markierte und nicht klassifizierte Daten und klassifizieren und markieren Sie die Daten entsprechend.
  - Ein Beispiel finden Sie unter [Data Catalog and crawlers in AWS Glue](#) (Datenkatalog und Crawler in AWS Glue).

- Richten Sie einen Datenkatalog ein, der Audit- und Governance-Funktionen bietet.
- Definieren und dokumentieren Sie Bearbeitungsverfahren für jede Datenklasse.
- Prüfen Sie mithilfe von Automatisierung die Umgebung regelmäßig auf nicht markierte und nicht klassifizierte Daten und klassifizieren und markieren Sie die Daten entsprechend.

## Ressourcen

### Zugehörige Dokumente:

- [Nutzung der AWS Cloud zur Unterstützung der Datenklassifizierung](#)
- [Tag-Richtlinien von AWS Organizations](#)

### Zugehörige Videos:

- [Enabling agility with data governance on AWS](#) (Mehr Agilität mit Data Governance auf AWS)

## SUS04-BP02 Verwenden von Technologien, die Datenzugriff und Speichermuster unterstützen

Nutzen Sie Speichertechnologien, die den Zugriff auf Ihre Daten und ihre Speicherung jeweils optimal unterstützen, um die Zahl der bereitgestellten Ressourcen zu minimieren und gleichzeitig den Workload zu unterstützen.

### Typische Anti-Muster:

- Sie gehen davon aus, dass für alle Workloads ähnliche Datenspeicher- und Zugriffsmuster gelten.
- Sie verwenden nur eine Speicherebene, vorausgesetzt, dass alle Workloads in diese Ebene passen.
- Sie gehen davon aus, dass Datenzugriffsmuster im Laufe der Zeit konsistent bleiben.

Vorteile der Nutzung dieser bewährten Methode: Die Auswahl und Optimierung Ihrer Speichertechnologien auf der Grundlage von Datenzugriffs- und Speichermustern hilft Ihnen, die erforderlichen Cloud-Ressourcen zu reduzieren, um Ihre Geschäftsanforderungen zu erfüllen und die Gesamteffizienz des Cloud-Workloads zu verbessern.

Risikostufe bei fehlender Befolgung dieser Best Practice: Niedrig



## Implementierungsleitfaden

Wählen Sie für maximale Leistungseffizienz die für Ihre Zugriffsmuster geeignete Speicherlösung, oder passen Sie Ihre Zugriffsmuster an die Speicherlösung an.

- Bewerten Sie Ihre Datenmerkmale und Zugriffsmuster, um die wichtigsten Merkmale Ihres Speicherbedarfs zu erfassen. Zu den berücksichtigenden Schlüsselmerkmalen gehören:
  - Datentyp: strukturiert, semistrukturiert, unstrukturiert
  - Datenwachstum: begrenzt, unbegrenzt
  - Stabilität von Daten: anhaltend, flüchtig, vorübergehend
  - Zugriffsmuster: Lese- oder Schreibvorgänge, Frequenz, Spitzen oder Konsistenz
- Migrieren Sie Daten auf die geeignete Speichertechnologie, die Ihre Datenmerkmale und Zugriffsmuster unterstützt. Hier sind einige Beispiele für AWS-Speichertechnologien und ihre Schlüsselmerkmale:

Typ	Technologie	Schlüsselmerkmale
Objektspeicher	<a href="#">Amazon S3</a>	Ein Objektspeicherservice mit unbegrenzter Skalierbarkeit, hoher Verfügbarkeit und mehreren Zugriffsoptionen. Für die Übertragung von Objekten in und aus Amazon S3 und den Zugriff auf diese Objekte können Sie einen Service wie z. B. <a href="#">Transfer Acceleration</a> oder <a href="#">Zugriffspunktenutzen</a> , um Ihren Standort, Ihre Sicherheitsanforderungen und Zugriffsmuster zu unterstützen.
Archivieren von Speichern	<a href="#">Amazon S3 Glacier</a>	Speicherklasse von Amazon S3 für die Datenarchivierung.

Typ	Technologie	Schlüsselmerkmale
Gemeinsames Dateisystem	<a href="#">Amazon Elastic File System (Amazon EFS)</a>	Mountfähiges Dateisystem, auf das verschiedene Arten von Datenverarbeitungs-lösungen zugreifen können. Amazon EFS erweitert und verringert den Speicher automatisch und ist leistungs-optimiert, um durchgängig niedrige Latenzen zu bieten.
Gemeinsames Dateisystem	<a href="#">Amazon FSx</a>	Basiert auf den neuesten AWS-Datenverarbeit-ungslösungen und unterstüt-zt vier gängige Dateisysteme: NetApp ONTAP, OpenZFS, Windows File Server und Lustre. Die Latenz, der Durchsatz und die IOPS von Amazon FSx <a href="#">variieren</a> je nach Dateisystem und sollten bei der Auswahl des richtigen Dateisystems für Ihre Workload-Anforderungen berücksichtigt werden.

Typ	Technologie	Schlüsselmerkmale
Blockspeicher	<a href="#">Amazon Elastic Block Store (Amazon EBS)</a>	Skalierbarer, hochleistungsfähiger Blockspeicherservice für Amazon Elastic Compute Cloud (Amazon EC2). Amazon EBS umfasst SSD-gestützten Speicher für transaktions- und IOPS-intensive Workloads und HDD-gestützten Speicher für durchsatzintensive Workloads.
Relationale Datenbank	<a href="#">Amazon Aurora</a> , <a href="#">Amazon RDS</a> , <a href="#">Amazon Redshift</a>	Sie unterstützt ACID-Transaktionen (Atomarität, Konsistenz, Isolation und Dauerhaftigkeit) und gewährleistet die referentielle Integrität sowie eine starke Datenkonsistenz. Bei zahlreichen herkömmlichen Anwendungen, Enterprise Resource Planning (ERP), Customer Relationship Management (CRM) und E-Commerce-Systemen werden relationale Datenbanken zum Speichern der Daten verwendet.

Typ	Technologie	Schlüsselmerkmale
Schlüssel-Werte-Datenbank	<a href="#">Amazon DynamoDB</a>	Für gängige Zugriffsmuster optimiert, üblicherweise zum Speichern und Abrufen großer Datenmengen. Web-Apps mit hohem Datenverkehr, E-Commerce-Systeme und Gaming-Anwendungen sind typische Anwendungsfälle für Schlüssel-Werte-Datenbanken.

- Bei Speichersystemen, die eine feste Größe haben, wie z. B. Amazon EBS oder Amazon FSx, überwachen Sie den verfügbaren Speicherplatz und automatisieren die Speicherzuweisung bei Erreichen eines Schwellenwertes. Sie können mithilfe von Amazon CloudWatch verschiedene Metriken für [Amazon EBS](#) und [Amazon FSx](#).
- Amazon S3-Speicherklassen können auf Objektebene konfiguriert werden und ein einzelner Bucket kann Objekte enthalten, die in allen Speicherklassen gespeichert sind.
- Sie können auch Amazon S3-Lebenszyklusrichtlinien verwenden, um Objekte automatisch zwischen Speicherklassen zu wechseln oder Daten zu entfernen, ohne dass die Anwendung geändert werden muss. Im Allgemeinen müssen Sie bei diesen Speichermechanismen einen Kompromiss zwischen Ressourceneffizienz, Zugriffslatenz und Zuverlässigkeit eingehen.

## Ressourcen

### Zugehörige Dokumente:

- [Amazon EBS-Volume-Typen](#)
- [Amazon EC2-Instance-Speicher](#)
- [Amazon S3 Intelligent-Tiering](#)
- [Amazon EBS -E/A-Merkmale](#)
- [Verwenden von Amazon S3-Speicherklassen](#)
- [Was ist Amazon S3 Glacier?](#)

### Zugehörige Videos:

- [Architectural Patterns for Data Lakes on AWS \(Architekturmodelle für Data Lakes in AWS\)](#)
- [Ausführliche Beschreibung von Amazon EBS \(STG303-R1\)](#)
- [Optimieren Sie Ihre Speicherleistung mit Amazon S3 \(STG343\)](#)
- [Building modern data architectures on AWS \(Erstellen von modernen Datenarchitekturen auf AWS\)](#)

### Zugehörige Beispiele:

- [Amazon EFS-CSI-Treiber](#)
- [Amazon EBS-CSI-Treiber](#)
- [Amazon EFS-Dienstprogramme](#)
- [Amazon EBS – automatische Skalierung](#)
- [Amazon S3-Beispiele](#)

## SUS04-BP03 Verwalten des Lebenszyklus von Datensätzen mithilfe von Richtlinien

Verwalten Sie den Lebenszyklus aller Daten und setzen Sie automatisch Löschen durch, um den für Ihren Workload benötigten Speicher insgesamt zu minimieren.

### Typische Anti-Muster:

- Sie löschen Daten manuell.
- Sie löschen keine Workload-Daten.
- Sie verschieben Daten nicht abhängig von den Aufbewahrungs- und Zugriffsanforderungen in energieeffizientere Speicherebenen.

Vorteile der Einführung dieser bewährten Methode: Durch Richtlinien für den Lebenszyklus wird die Effizienz des Datenzugriffs und der Datenaufbewahrung für einen Workload sichergestellt.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: Mittel

## Implementierungsleitfaden

Datensätze verfügen während ihres Lebenszyklus normalerweise über unterschiedliche Aufbewahrungs- und Zugriffsanforderungen. So kann eine Anwendung z. B. für einen bestimmten Zeitraum häufig Zugriff auf einige Datensätze benötigen. Danach wird nur noch unregelmäßig darauf zugegriffen.

Um Datensätze während ihres Lebenszyklus effizient zu verwalten, konfigurieren Sie Lebenszyklusrichtlinien, d. h. Regeln, die den Umgang mit den Datensätzen definieren.

Mit Lebenszyklus-Konfigurationsregeln können Sie einen bestimmten Speicherservice anweisen, einen Datensatz in energieeffizientere Speicherebenen zu verschieben, ihn zu archivieren oder zu löschen.

### Implementierungsschritte

- [Klassifizieren Sie die Datensätze in Ihrem Workload.](#)
- Definieren Sie Bearbeitungsverfahren für jede Datenklasse.
- Legen Sie automatisierte Lebenszyklusrichtlinien zur Durchsetzung von Lebenszyklusregeln fest. Hier finden Sie einige Beispiel zum Einrichten von automatisierten Lebenszyklusrichtlinien für unterschiedliche AWS-Speicherservices:

Storage service	How to set automated lifecycle policies
<a href="#">Amazon S3</a>	<p>Mit <a href="#">Amazon S3-Lebenszyklen</a> können Sie Ihre Objekte während ihres gesamten Lebenszyklus verwalten. Wenn die Zugriffsmuster unbekannt oder nicht prognostizierbar sind oder sich ändern, können Sie <a href="#">Amazon S3 Intelligent-Tiering</a> verwenden. Hiermit werden Zugriffsmuster überwacht und Objekte, auf die nicht zugegriffen wurde, automatisch in kostengünstigere Zugriffsebenen verschoben. Anhand von <a href="#">Amazon S3 Storage Lens</a>-Metriken können Sie Optimierungsmöglichkeiten und Lücken im Lebenszyklusmanagement ermitteln.</p>

Storage service	How to set automated lifecycle policies
<a href="#">Amazon Elastic Block Store</a>	Mit <a href="#">Amazon Data Lifecycle Manager</a> lassen sich Erstellen, Aufbewahrung und Löschen von Amazon EBS-Snapshots und Amazon EBS-gestützten AMIs automatisieren.
<a href="#">Amazon Elastic File System</a>	Das <a href="#">Amazon EFS-Lebenszyklusmanagement</a> verwaltet den Dateispeicher für Ihre Dateisysteme automatisch.
<a href="#">Amazon Elastic Container Registry</a>	<a href="#">Amazon ECR-Lebenszyklusrichtlinien</a> automatisieren die Bereinigung von Container-Images, indem Images abhängig von Alter oder Anzahl ablaufen.
<a href="#">AWS Elemental MediaStore</a>	Sie können eine <a href="#">Objektlebenszyklus-Richtlinie</a> verwenden, die steuert, wie lange Objekte im MediaStore-Container gespeichert werden sollen.

- Löschen Sie nicht genutzte Volumes, Snapshots und Daten, deren Aufbewahrungszeitraum abgelaufen ist. Nutzen Sie zum Löschen native Servicefunktionen wie Amazon DynamoDB Time To Live oder die Amazon CloudWatch-Protokollaufbewahrung.
- Aggregieren und komprimieren Sie Daten wenn möglich auf der Basis von Lebenszyklusregeln.

## Ressourcen

### Zugehörige Dokumente:

- [Optimize your Amazon S3 Lifecycle rules with Amazon S3 Storage Class Analysis](#) (Optimieren von S3-Lebenszyklusregeln mit S3 Storage Class Analysis)
- [Evaluating Resources with AWS-Config-Regeln](#) (Evaluieren von Ressourcen mit AWS Config-Regeln)

### Zugehörige Videos:

- [Simplify Your Data Lifecycle and Optimize Storage Costs With Amazon S3 Lifecycle](#) (Vereinfachen des Datenlebenszyklus und Optimieren von Speicherkosten mit S3-Lebenszyklen)
- [Reduce Your Storage Costs Using Amazon S3 Storage Lens](#) (Reduzieren von Speicherkosten mit S3 Storage Lens)

## SUS04-BP04 Verwendung von Elastizität und Automatisierung zur Erweiterung des Block-Speichers oder des Dateisystems

Verwenden Sie Elastizität und Automatisierung, um den Block-Speicher oder das Dateisystem zu erweitern, wenn das Datenvolumen zunimmt, um den bereitgestellten Gesamtspeicher zu minimieren.

Typische Anti-Muster:

- Sie unterhalten einen großen Block-Speicher oder ein großes Dateisystem für künftige Anforderungen.
- Sie stellen zu viele Input- und Output-Operationen pro Sekunde (IOPS) in Ihrem Dateisystem bereit.
- Sie überwachen die Nutzung Ihrer Daten-Volumes nicht.

Vorteile der Nutzung dieser bewährten Methode: Die Minimierung der übermäßigen Bereitstellung für das Speichersystem reduziert ungenutzte Ressourcen und verbessert die Gesamteffizienz Ihres Workloads.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: Mittel

### Implementierungsleitfaden

Erstellen Sie Block-Speicher und Dateisysteme mit Größenzuweisung, Durchsatz und Latenz, die den Anforderungen Ihres Workloads entsprechen. Verwenden Sie Elastizität und Automatisierung, um den Block-Speicher oder das Dateisystem zu erweitern, wenn das Datenvolumen zunimmt, ohne dass diese Speicherservices übermäßig bereitgestellt werden.

### Implementierungsschritte

- Stellen Sie bei Speichersystemen mit einer festen Größe wie [Amazon EBS](#) sicher, dass Sie die Menge des verwendeten Speichers im Vergleich zur Gesamtspeichergöße überwachen und nach Möglichkeit die Speichergröße beim Erreichen eines Schwellenwerts automatisch erhöhen.



- Verwenden Sie elastische Volumes und verwaltete Blockdaten-Services, um automatisch zusätzlichen Speicher zuzuweisen, wenn die Menge der persistenten Daten wächst. Sie können beispielsweise [Amazon EBS Elastic Volumes](#) verwenden, um Volume-Größe, Volume-Typ oder die Leistung Ihrer Amazon EBS-Volumes zu modifizieren.
- Wählen Sie die korrekte Speicherklasse sowie den korrekten Leistungs- und Durchsatz-Modus für Ihr Dateisystem für Ihre geschäftlichen Anforderungen und überschreiten Sie diese nicht.
  - [Amazon EFS Leistung](#)
  - [Amazon EBS-Volume-Leistung auf Linux-Instances](#)
- Legen Sie Zielstufen für die Nutzung Ihrer Daten-Volumes fest und passen Sie die Größe von Volumes an, die außerhalb der erwarteten Bereiche liegen.
- Passen Sie die Größe schreibgeschützter Volumes an die Datenmenge an.
- Migrieren Sie Daten zu Objektspeichern, um zu vermeiden, dass die überschüssige Kapazität aus Volumes mit fester Größe im Blockspeicher bereitgestellt wird.
- Überprüfen Sie elastische Volumes und Dateisysteme, beenden Sie nicht genutzte und verkleinern Sie zu große Volumes, um sie an den aktuellen Datenumfang anzupassen.

## Ressourcen

### Zugehörige Dokumente:

- [Amazon FSx-Dokumentation](#)
- [Was ist Amazon Elastic File System?](#)

### Zugehörige Videos:

- [Deep Dive on Amazon EBS Elastic Volumes](#) (Weiterführende Informationen zu Amazon EBS Elastic Volumes)
- [Amazon EBS and Snapshot Optimization Strategies for Better Performance and Cost Savings](#) (Amazon EBS und Snapshot-Optimierungsstrategien für bessere Leistung und Kosteneinsparungen)
- [Optimizing Amazon EFS for cost and performance, using best practices](#) (Amazon EFS mithilfe bewährter Methoden für Kosten und Leistung optimieren)

## SUS04-BP05 Entfernen nicht benötigter oder redundanter Daten

Entfernen Sie nicht benötigte oder redundante Daten, um die zum Speichern Ihrer Datensätze benötigten Speicherressourcen zu minimieren.

Typische Anti-Muster:

- Sie duplizieren Daten, die leicht abgerufen oder erneut erstellt werden können.
- Sie sichern alle Daten, ohne ihre Kritikalität zu berücksichtigen.
- Sie löschen Daten nur unregelmäßig, nur bei bestimmten Ereignissen oder gar nicht.
- Sie speichern Daten redundant, unabhängig von der Stabilität des Speicherservices.
- Sie aktivieren die Amazon S3-Versionsverwaltung, ohne dass dies geschäftlich gerechtfertigt ist.

Vorteile der Einführung dieser bewährten Methode: Durch das Entfernen nicht benötigter Daten werden die für Ihren Workload benötigte Speichergröße und die Umweltbelastungen durch den Workload reduziert.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: Mittel

### Implementierungsleitfaden

Speichern Sie keine Daten, die Sie nicht benötigen. Automatisieren Sie das Löschen von nicht benötigten Daten. Verwenden Sie Technologien, die Daten auf Datei- und Blockebene deduplizieren. Nutzen Sie native Servicefunktionen für Replikation und Redundanz.

### Implementierungsschritte

- Bewerten Sie, ob Sie das Speichern von Daten vermeiden können, indem Sie vorhandene, öffentlich verfügbare Datensätze in [AWS Data Exchange](#) und [offene Daten in AWS](#) verwenden.
- Verwenden Sie Mechanismen, die Daten auf Block- und Objektebene deduplizieren können. Hier finden Sie einige Beispiele zum Deduplizieren von Daten in AWS:

Storage service	Deduplication mechanism
<a href="#">Amazon S3</a>	Verwenden Sie <a href="#">AWS Lake Formation FindMatches</a> und das neue FindMatches ML Transform, um übereinstimmende Einträge in

Storage service	Deduplication mechanism
	einem Datensatz zu finden (auch solche ohne ID).
<a href="#">Amazon FSx</a>	Aktivieren Sie die <a href="#">Datendeduplizierung</a> in Amazon FSx für Windows.
<a href="#">Amazon Elastic Block Store-Snapshots</a>	Bei Snapshots handelt es sich um inkrementelle Sicherungen. Das bedeutet, dass nur die Blöcke auf dem Gerät gespeichert werden, die sich seit dem letzten Snapshot geändert haben.

- Analysieren Sie den Datenzugriff, um nicht benötigte Daten zu identifizieren. Automatisieren Sie Lebenszyklusrichtlinien. Nutzen Sie zum Löschen native Servicefunktionen wie [Amazon DynamoDB Time To Live](#), [Amazon S3-Lebenszyklen](#) oder die [Amazon CloudWatch-Protokollaufbewahrung](#).
- Verwenden Sie Virtualisierungsfunktionen in AWS, um Daten an der Quelle beizubehalten und eine Duplikation zu vermeiden.
  - [Cloud Native Data Virtualization on AWS](#) (Cloudnative Datenvirtualisierung in AWS)
  - [Lab: Optimize Data Pattern Using Amazon Redshift Data Sharing](#) (Lab: Optimierung von Datenmustern mit Amazon Redshift Data Sharing)
- Verwenden Sie Sicherungstechnologien, mit denen inkrementelle Sicherungen möglich sind.
- Nutzen Sie zum Erfüllen der Stabilitätsziele die Stabilität von [Amazon S3](#) und [Replikation von Amazon EBS](#) anstelle von selbst verwalteten Technologien wie redundanten Arrays unabhängiger Datenträger (Redundant Array Of Independent Disks, RAID).
- Zentralisieren Sie Protokoll- und Nachverfolgungsdaten, deduplizieren Sie identische Protokolleinträge und richten Sie Mechanismen für die Anpassung der Ausführlichkeit ein, wenn notwendig.
- Füllen Sie Zwischenspeicher nur vorab aus, wenn dies begründet werden kann.
- Richten Sie Überwachung und Automatisierung für den Cache ein, um seine Größe entsprechend anzupassen.
- Entfernen Sie veraltete Bereitstellungen und Komponenten aus Objektspeichern und Edge-Zwischenspeichern, wenn Sie neue Versionen Ihres Workloads veröffentlichen.

## Ressourcen

### Zugehörige Dokumente:

- [Change log data retention in CloudWatch Logs](#) (Ändern der Protokolldatenaufbewahrung in CloudWatch Logs)
- [Data deduplication on Amazon FSx for Windows File Server](#) (Dateneduplizierung in Amazon FSx für Windows File Server)
- [Features of Amazon FSx for ONTAP including data deduplication](#) (Funktionen von Amazon FSx for ONTAP einschließlich Dateneduplizierung)
- [Invalidating Files on Amazon CloudFront](#) (Invalidieren von Dateien auf Amazon CloudFront)
- [Using AWS Backup to back up and restore Amazon EFS file systems](#) (Verwenden von AWS Backup, um Amazon EFS-Dateisysteme zu sichern und wiederherzustellen)
- [Was ist Amazon CloudWatch Logs?](#)
- [Working with backups on Amazon RDS](#) (Arbeiten mit Backups in RDS)

### Zugehörige Videos:

- [Fuzzy Matching and Deduplicating Data with ML Transforms for AWS Lake Formation](#) (Fuzzy Matching und Deduplizieren von Daten mit ML Transforms für AWS Lake Formation)

### Zugehörige Beispiele:

- [Wie analysiere ich meine Amazon S3-Serverzugriffsprotokolle mit Amazon Athena?](#)

## SUS04-BP06 Verwenden geteilter Dateisysteme oder Objektspeicher für den Zugriff auf allgemeine Daten

Verwenden Sie geteilte Dateisysteme oder Speicher, um Datenduplizierungen zu vermeiden und eine effizientere Infrastruktur für Ihren Workload zu ermöglichen.

### Typische Anti-Muster:

- Sie stellen für jeden einzelnen Client Speicher bereit.
- Sie trennen Datenvolumina von inaktiven Clients nicht ab.
- Sie ermöglichen keinen Zugriff auf Speicher über Plattformen und Systeme hinweg.

Vorteile der Nutzung dieser bewährten Methode: Die Verwendung geteilter Dateisysteme oder Speicher ermöglicht die gemeinsame Nutzung von Daten für mehrere Nutzer, ohne dass diese dazu kopiert werden müssen. Dies reduziert den Ressourcenumfang für den Workload.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: Mittel

## Implementierungsleitfaden

Wenn Sie mehrere Nutzer oder Anwendungen haben, die auf die gleichen Datensätze zugreifen müssen, ist die Verwendung geteilter Speichertechnologien wichtig für eine effiziente Infrastruktur für Ihren Workload. Solche Technologien bieten einen zentralen Speicherort für die Speicherung und Verwaltung von Datensätzen und zur Vermeidung von Datenduplizierungen. Dazu wird die Konsistenz der Daten über verschiedene Systeme hinweg durchgesetzt. Dazu kommt, dass geteilte Speicher die effizientere Nutzung der Computing-Kapazitäten ermöglichen, da mehr Computing-Ressourcen gleichzeitig auf Daten zugreifen und diese verarbeiten können.

Rufen Sie Daten von diesen geteilten Speicherservices nur bei Bedarf ab und trennen Sie nicht genutzte Volumes, um Ressourcen freizumachen.

## Implementierungsschritte

- Migrieren Sie Daten in einen geteilten Speicher, wenn die Daten mehrfach genutzt werden. Hier sind einige Beispiele für geteilte Speichertechnologien auf AWS:

Storage option	When to use
<a href="#">Amazon EBS Multi-Attach</a>	Amazon EBS Multi-Attach ermöglicht die Anfügung eines einzelnen Provisioned IOPS SSD (io1 oder io2)-Volumes an mehrere Instances in derselben Availability Zone.
<a href="#">Amazon EFS</a>	Vgl. <a href="#">Auswahl von Amazon EFS</a> .
<a href="#">Amazon FSx</a>	Vgl. <a href="#">Auswahl eines Amazon FSx-Datei systems</a> .
<a href="#">Amazon S3</a>	Anwendungen, die keine Dateisystemstruktur benötigen und zur Arbeit mit Objektspeichern gedacht sind, können Amazon S3

Storage option	When to use
	als massive, skalierbare, dauerhafte und kostengünstige Speicherlösung nutzen.

- Kopieren Sie Daten bzw. rufen Sie sie nur dann von geteilten Dateisystemen ab, wenn Sie sie benötigen. Sie können beispielsweise ein [Amazon FSx for Lustre-Dateisystem mit Unterstützung durch Amazon S3](#) erstellen und nur die Teilmenge der Daten laden, die für die Verarbeitung von Aufgaben zu Amazon FSx benötigt werden.
- Löschen Sie Daten entsprechend Ihren Nutzungsmustern, wie in [SUS04-BP03 Verwalten des Lebenszyklus von Datensätzen mithilfe von Richtlinien](#) erläutert.
- Trennen Sie Volumes von Clients, die sie nicht aktiv verwenden.

## Ressourcen

### Zugehörige Dokumente:

- [Linking your file system to an Amazon S3 bucket](#) (Verknüpfung Ihres Dateisystems mit einem Amazon S3-Bucket)
- [Using Amazon EFS for AWS Lambda in your serverless applications](#) (Amazon EFS für AWS Lambda in Ihren Serverless-Anwendungen verwenden)
- [Amazon EFS Intelligent-Tiering Optimizes Costs for Workloads with Changing Access Patterns](#) (Amazon EFS Intelligent-Tiering optimiert die Kosten für Workloads mit wechselnden Zugriffsmustern)
- [Using Amazon FSx with your on-premises data repository](#) (Verwendung von Amazon FSx mit Ihrem On-Premises-Daten-Repository)

### Zugehörige Videos:

- [Optimierung der Speicherkosten mit Amazon EFS](#) (Optimierung der Speicherkosten mit Amazon EFS)

## SUS04-BP07 Minimieren von Datenübertragungen zwischen Netzwerken

Verwenden Sie gemeinsam genutzte Dateisysteme oder Objektspeicher zum Zugriff auf häufig genutzte Daten und minimieren Sie die zur Unterstützung von Datenverschiebungen für Ihren Workload benötigten Netzwerkressourcen.

Typische Anti-Muster:

- Sie speichern alle Daten in derselben AWS-Region, unabhängig davon, wo sich deren Benutzer befinden.
- Sie optimieren Datenumfang und -format nicht vor der Verschiebung über das Netzwerk.

Vorteile der Nutzung dieser bewährten Methode: Die Optimierung der Datenverschiebung über das Netzwerk reduziert den Umfang der für den Workload benötigten Netzwerkressourcen und verringert die Umweltauswirkungen.

Risikostufe bei fehlender Befolgung dieser Best Practice: Mittel

### Implementierungsleitfaden

Das Verschieben von Daten in der gesamten Organisation erfordert Computing-, Netzwerk- und Speicherressourcen. Verwenden Sie Techniken zur Minimierung von Datenverschiebungen und verbessern Sie die Gesamteffizienz Ihres Workloads.

### Implementierungsschritte

- Berücksichtigen Sie die Nähe zu den Daten oder Benutzern als Entscheidungsfaktor bei der [Auswahl einer Region für Ihren Workload](#).
- Partitionieren Sie regional genutzte Services so, dass regionsspezifische Daten in der Region gespeichert werden, in der sie genutzt werden.
- Verwenden Sie effiziente Dateiformate (wie etwa Parquet oder ORC) und komprimieren Sie die Daten, bevor Sie sie über das Netzwerk verschieben.
- Verschieben Sie keine nicht genutzten Daten. Einige Beispiele, die Ihnen helfen können, das Verschieben ungenutzter Daten zu vermeiden:
  - Beschränken Sie API-Antworten nur auf relevante Daten.
  - Aggregieren Sie Daten, wenn keine detaillierten Informationen auf Datensatzebene benötigt werden.
- Siehe [Well-Architected Lab – Optimierung von Datenmustern mit Amazon Redshift Data Sharing](#).

- Erwägen Sie die [kontoübergreifende Datenfreigabe in AWS Lake Formation](#).
- Nutzen Sie Services, die Ihnen dabei helfen können, Code näher an den Nutzern Ihres Workloads auszuführen:

Service	Verwendung
<a href="#">Lambda@Edge</a>	Verwenden Sie dies für rechenintensive Anwendungen, die ausgeführt werden, wenn sich Objekte nicht im Zwischenspeicher befinden.
<a href="#">CloudFront-Funktionen</a>	Verwenden Sie diese für einfache Anwendungsfälle wie HTTP(s)-Anfragen oder Antwortmanipulationen, die von kurzlebigen Funktionen initiiert werden können.
<a href="#">AWS IoT Greengrass</a>	Führen Sie lokale Rechenoperationen, Messaging sowie die Datenzwischenspeicherung für verbundene Geräte aus.

## Ressourcen

### Zugehörige Dokumente:

- [Optimieren Ihrer AWS-Infrastruktur für Nachhaltigkeit, Teil III: Netzwerke](#)
- [Globale AWS-Infrastruktur](#)
- [Hauptfunktionen von Amazon CloudFront einschließlich CloudFront Globales Edge-Netzwerk](#)
- [Komprimieren von HTTP-Anforderungen in Amazon OpenSearch Service](#)
- [Zwischenkomprimierung von Daten mit Amazon EMR](#)
- [Laden komprimierter Datendateien aus Amazon S3 in Amazon Redshift](#)
- [Bereitstellen von komprimierten Dateien mit Amazon CloudFront](#)

### Zugehörige Videos:

- [Demystifying data transfer on AWS \(Das Geheimnis der Datenübertragung in AWS lüften\)](#)



Zugehörige Beispiele:

- [Nachhaltige Architektur — Minimierung des Datenverkehrs zwischen Netzwerken](#)

## SUS04-BP08 Sichern von Daten nur in dem Fall, wenn ihre erneute Erstellung schwierig ist

Vermeiden Sie das Sichern von Daten ohne geschäftlichen Wert, um die Anforderungen an Speicherressourcen für Ihren Workload zu minimieren.

Typische Anti-Muster:

- Sie haben keine Sicherungsstrategie für Ihre Daten.
- Sie sichern Daten, die problemlos erneut erstellt werden können.

Vorteile der Nutzung dieser bewährten Methode: Das Vermeiden der Sicherung nichtkritischer Daten reduziert den Umfang der benötigten Speicherressourcen für den Workload und verringert die Umweltauswirkungen.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: Mittel

### Implementierungsleitfaden

Die Vermeidung der Sicherung nicht benötigter Daten kann Kosten senken und die von dem Workload verwendeten Speicherressourcen verringern. Sichern Sie nur Daten, die einen geschäftlichen Wert haben oder zur Erfüllung von Compliance-Anforderungen benötigt werden. Prüfen Sie Backup-Richtlinien und vermeiden Sie einen flüchtigen Speicher, der in einem Wiederherstellungsszenario keinen Wert bietet.

Implementierungsschritte

- Implementieren Sie eine Richtlinie für die Klassifizierung von Daten wie in [SUS04-BP01 Implementieren einer Richtlinie für die Klassifizierung von Daten](#) erläutert.
- Nutzen Sie die Wichtigkeit Ihrer Datenklassifizierung und entwerfen Sie eine Sicherungsstrategie auf der Grundlage Ihrer [Recovery Time Objective \(RTO\)](#) und Ihrer [Recovery Point Objective \(RPO\)](#). Vermeiden Sie die Sicherung nichtkritischer Daten.
  - Schließen Sie Daten aus, die problemlos erneut erstellt werden können.
  - Schließen Sie flüchtige Daten von Backups aus.

- Schließen Sie lokale Kopien von Daten aus, es sei denn, die für die Wiederherstellung dieser Daten von einem gemeinsamen Standort benötigte Zeit überschreitet Ihre Service Level Agreements (SLAs).
- Verwenden Sie eine automatisierte Lösung oder einen verwalteten Service zur Sicherung geschäftskritischer Daten.
- [AWS Backup](#) ist ein vollständig verwalteter Service, der die Zentralisierung und Automatisierung des Schutzes von Daten für AWS-Services in der Cloud und On-Premises vereinfacht. Praktische Anleitungen zur Erstellung automatisierter Sicherungen mit AWS Backup finden Sie unter [Well-Architected Labs – Testen der Sicherung und Wiederherstellung Ihrer Daten](#).
- [Automatisieren Sie Sicherungen und optimieren Sie die Sicherungskosten für Amazon EFS mit AWS Backup](#).

## Ressourcen

Zugehörige bewährte Methoden:

- [REL09-BP01 Ermitteln und Sichern aller zu sichernden Daten oder Reproduzieren der Daten aus Quellen](#)
- [REL09-BP03 Automatische Daten-Backups](#)
- [REL13-BP02: Verwenden von definierten Wiederherstellungsstrategien, um die Wiederherstellungsziele zu erreichen](#)

Zugehörige Dokumente:

- [Using AWS Backup to back up and restore Amazon EFS file systems](#) (Verwenden von AWS Backup, um Amazon EFS-Dateisysteme zu sichern und wiederherzustellen)
- [Amazon EBS-Snapshots](#)
- [Arbeiten mit Backups in Amazon Relational Database Service](#)
- [APN-Partner: Partner, die Sie bei der Sicherung unterstützen können](#)
- [AWS Marketplace: Für die Sicherung geeignete Produkte](#)
- [Backing Up Amazon EFS](#) (Sichern von Amazon EFS)
- [Backing Up Amazon FSx for Windows File Server](#) (Sichern von Amazon FSx für Windows File Server)
- [Backup und Wiederherstellung für Amazon ElastiCache for Redis](#)

## Zugehörige Videos:

- [AWS re:Invent 2021 - Backup, disaster recovery, and ransomware protection with AWS](#) (AWS re:Invent 2021 – Backup, Notfallwiederherstellung und Ransomware-Schutz mit AWS)
- [AWS Backup Demo: Cross-Account and Cross-Region Backup](#) (AWS Backup-Demo: Konto- und regionsübergreifendes Backup)
- [AWS re:Invent 2019: Deep dive on AWS Backup, ft. Rackspace \(STG341\)](#) (AWS re:Invent 2019: Eingehende Informationen zu AWS Backup mit Rackspace (STG341))

## Zugehörige Beispiele:

- [Well-Architected Lab - Testing Backup and Restore of Data](#) (Well-Architected Lab – Testen von Backup und Wiederherstellung von Daten)
- [Well-Architected Lab - Backup and Restore with Failback for Analytics Workload](#) (Well-Architected Lab – Backups und Wiederherstellung mit Failback für Analytics-Workload )
- [Well-Architected Lab - Disaster Recovery - Backup and Restore](#) (Well-Architected Lab – Notfallwiederherstellung – Backup und Wiederherstellung)

# Hardware und Services

Suchen Sie nach Möglichkeiten, die Auswirkungen auf die Nachhaltigkeit Ihrer Workloads durch Änderungen der Methoden für die Hardwareverwaltung zu reduzieren. Minimieren Sie den Umfang der für die Bereitstellung erforderlichen Hardware und wählen Sie die jeweils effizienteste Hardware und den effizientesten Service für den jeweiligen Workload aus.

## Bewährte Methoden

- [SUS05-BP01 Verwenden der geringstmöglichen Menge an Hardware zur Erfüllung Ihrer Anforderungen](#)
- [SUS05-BP02 Verwenden von Instance-Typen mit den geringsten Auswirkungen](#)
- [SUS05-BP03 Verwenden verwalteter Services](#)
- [SUS05-BP04 Optimieren der Nutzung von hardwarebasierten Computing-Beschleunigern](#)

## SUS05-BP01 Verwenden der geringstmöglichen Menge an Hardware zur Erfüllung Ihrer Anforderungen

Verwenden Sie die geringstmögliche Menge an Hardware für Ihr Workload, um Ihre geschäftlichen Anforderungen in effizienter Weise zu erfüllen.

Typische Anti-Muster:

- Sie überwachen die Ressourcenauslastung nicht.
- Sie haben Ressourcen mit geringer Auslastung in Ihrer Architektur.
- Sie prüfen die Nutzung statischer Hardware nicht, um festzustellen, ob sie neu dimensioniert werden muss.
- Sie formulieren keine Ziele für die Hardwarenutzung in Ihrer Computing-Infrastruktur auf der Grundlage geschäftlicher KPIs.

Vorteile der Nutzung dieser bewährten Methode: Die korrekte Dimensionierung Ihrer Cloud-Ressourcen hilft dabei, die Umweltauswirkungen von Workloads zu reduzieren, Geld zu sparen und Leistungsbenchmarks einzuhalten.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: Mittel

### Implementierungsleitfaden

Wählen Sie die optimale Anzahl von Hardwaregeräten für Ihren Workload aus, um die allgemeine Effizienz zu verbessern. AWS Cloud bietet die Flexibilität, Ressourcen dynamisch durch verschiedene Mechanismen wie etwa [AWS Auto Scaling](#) zu erweitern oder zu reduzieren, um einem veränderten Bedarf gerecht zu werden. Dazu kommen [APIs und SDKs](#), mit denen Ressourcen mit minimalem Aufwand angepasst werden können. Verwenden Sie diese Möglichkeiten für häufige Änderungen an Ihren Workload-Implementierungen. Verwenden Sie dazu Dimensionierungsanleitungen von AWS-Tools für den effizienten Betrieb Ihrer Cloud-Ressourcen und die Erfüllung Ihrer geschäftlichen Anforderungen.

### Implementierungsschritte

- Wählen Sie die Instances, die am besten zu Ihren Anforderungen passen.
  - [How do I choose the appropriate Amazon EC2 instance type for my workload?](#) (Wie wähle ich einen geeigneten Amazon EC2-Instance-Typ für meinen Workload aus?)

- [Attribute based Instance Type Selection for Amazon EC2 for Fleet](#) (Attributbasierte Auswahl des Instance-Typs für die Amazon EC2-Fleet).
- [Erstellen Sie eine Auto Scaling-Gruppe unter Verwendung einer attributbasierten Auswahl des Instance-Typs.](#)
- Skalieren Sie für variable Workloads in kleinen Schritten.
- Verwenden Sie mehrere Computing-Einkaufsoptionen, um die Instance-Flexibilität, die Skalierbarkeit und Kosteneinsparungen ins Gleichgewicht zu bringen.
  - [On-Demand-Instances](#) eignen sich am besten für neue, statusbehaftete Workloads mit Spitzen, die hinsichtlich Instance-Typ, Standort oder Zeit nicht flexibel sein können.
  - [Spot Instances](#) eignen sich hervorragend zur Ergänzung der anderen Optionen für Anwendungen, die fehlertolerant und flexibel sind.
  - Nutzen Sie [Compute Savings Plans](#) für stabile Workloads, die Flexibilität ermöglichen, wenn sich Ihre Anforderungen (wie etwa AZ, Region, Instance-Familien oder Instance-Typen) ändern.
- Verwenden Sie unterschiedliche Instances und Availability Zones zur Maximierung der Anwendungsverfügbarkeit und nutzen Sie nach Möglichkeit überschüssige Kapazität.
- Verwenden Sie die Empfehlungen zur Dimensionierung in AWS-Tools, um Anpassungen an Ihrem Workload vorzunehmen.
  - [AWS Compute Optimizer](#)
  - [AWS Trusted Advisor](#)
- Verhandeln Sie SLAs (Service Level Agreements), die eine vorübergehende Reduzierung von Kapazitäten zulassen, während die Bereitstellung von Ersatzressourcen automatisiert wird.

## Ressourcen

### Zugehörige Dokumente:

- [Optimizing your AWS Infrastructure for Sustainability, Part I: Compute](#) (Optimieren Ihrer AWS-Infrastruktur für Nachhaltigkeit, Teil I: Datenverarbeitung)
- [Attribute based Instance Type Selection for Auto Scaling for Amazon EC2 Fleet](#) (Attributbasierte Auswahl des Instance-Typs für Auto Scaling und die Amazon EC2 Fleet)
- [AWS Compute Optimizer-Dokumentation](#)
- [Operating Lambda: Performance optimization](#) (Ausführen von Lambda: Leistungsoptimierung)
- [Auto Scaling Documentation](#) (Dokumentation zu Auto Scaling)

## Zugehörige Videos:

- [Build a cost-, energy-, and resource-efficient compute environment](#) (Entwickeln einer kosten-, energie- und ressourceneffizienten Datenverarbeitungsumgebung)

## Zugehörige Beispiele:

- [Well-Architected Lab – Rightsizing with AWS Compute Optimizer and Memory Utilization Enabled \(Level 200\)](#) (Well-Architected Lab – Größenanpassung, wenn AWS Compute Optimizer und Speicherauslastung aktiviert sind (Stufe 200))

## SUS05-BP02 Verwenden von Instance-Typen mit den geringsten Auswirkungen

Überwachen und nutzen Sie kontinuierlich neue Instance-Typen, um Verbesserungen bei der Energieeffizienz zu nutzen.

### Typische Anti-Muster:

- Sie verwenden lediglich eine Familie von Instances.
- Sie verwenden nur x86-Instances.
- Sie geben einen Instance-Typ in Ihrer Amazon EC2 Auto Scaling-Konfiguration an.
- Sie verwenden AWS-Instances in einer Weise, für die sie nicht gedacht sind (beispielsweise Computing-optimierte Instances für speicherintensive Workloads).
- Sie evaluieren nicht regelmäßig die Instance-Typen.
- Sie prüfen nicht die Empfehlungen von AWS-Dimensionierungstools wie etwa [AWS Compute Optimizer](#).

Vorteile der Nutzung dieser bewährten Methode: Durch die Verwendung energieeffizienter und korrekt dimensionierter Instances können Sie die Umweltauswirkungen und die Kosten Ihrer Workloads deutlich reduzieren.

Risikostufe bei fehlender Befolgung dieser Best Practice: Mittel

## Implementierungsleitfaden

Die Verwendung effizienter Instances für Cloud-Workloads ist von entscheidender Bedeutung für eine geringere Ressourcennutzung und die Kosteneffizienz. Überwachen Sie kontinuierlich die Einführung neuer Instance-Typen und nutzen Sie Verbesserungen bei der Energieeffizienz, einschließlich Instance-Typen, die zur Unterstützung spezifischer Workloads bestimmt sind, wie z. B. Machine-Learning-Trainings und -Inferenzen und Videotranskodierung.

## Implementierungsschritte

- Informieren Sie sich über Instance-Typen, die die Umweltauswirkungen Ihrer Workloads reduzieren können.
  - Abonnieren Sie [Neuerungen bei AWS](#), um bei den neuesten AWS-Technologien und -Instances auf dem Laufenden zu bleiben.
  - Informieren Sie sich über die verschiedenen AWS-Instance-Typen.
  - Informieren Sie sich über auf AWS Graviton basierende Instances, die die höchste Leistung pro Watt in Amazon EC2 bieten; sehen Sie sich dazu Folgendes an: [re:Invent 2020 - Deep dive on AWS Graviton2 processor-powered Amazon EC2 instances \(Ein tiefer Einblick in vom AWS-Graviton2-Prozessor unterstützte EC2-Instances\)](#) und [Deep dive into AWS Graviton3 and Amazon EC2 C7g instances \(Ein tiefer Einblick in AWS-Graviton3- und EC2-C7g-Instances\)](#).
- Planen und übertragen Sie Ihre Workloads auf Instance-Typen mit den geringsten Auswirkungen.
  - Definieren Sie einen Prozess zur Evaluierung neuer Funktionen oder Instances für Ihre Workloads. Nutzen Sie die Agilität in der Cloud, um schnell zu testen, wie neue Instance-Typen die ökologische Nachhaltigkeit Ihrer Workloads verbessern können. Nutzen Sie Proxy-Metriken, um zu messen, wie viele Ressourcen Sie für eine Arbeitseinheit benötigen.
  - Modifizieren Sie Ihren Workload nach Möglichkeit so, dass er mit unterschiedlichen Zahlen von vCPUs und Arbeitsspeichergrößen kompatibel ist, um die größtmögliche Auswahl an Instance-Typen zu erhalten.
  - Erwägen Sie die Übertragung Ihres Workloads zu auf Graviton basierenden Instances, um die Leistungseffizienz Ihres Workloads zu verbessern.
    - [AWS Graviton-Schnellstart](#)
    - [Überlegungen bei der Übertragung von Workloads zu auf AWS Graviton basierenden Amazon Elastic Compute Cloud-Instances](#)
    - [AWS Graviton2 für ISVs](#)

- Erwägen Sie die Auswahl der AWS-Graviton-Option bei Ihrer Verwendung der [verwalteten AWS-Services](#).
- Migrieren Sie Ihren Workload zu Regionen mit Instances, die die geringsten nachhaltigkeitsbezogenen Auswirkungen bieten und dennoch Ihre geschäftlichen Anforderungen erfüllen.
- Nutzen Sie für Machine-Learning-Workloads spezielle Hardware, die auf Ihren Workload abgestimmt ist, z. B. [AWS Trainium](#), [AWS Inferentia](#) und [Amazon EC2 DL1](#). AWS Inferentia-Instances wie Inf2-Instances bieten eine um bis zu 50 % bessere Leistung pro Watt als vergleichbare Amazon EC2-Instances.
- Verwenden Sie [Amazon SageMaker Inference Recommender](#) für die Dimensionierung des ML-Inferenz-Endpunkts.
- Verwenden Sie für Workloads, bei denen es gelegentlich zu zusätzlichen Kapazitätsanforderungen kommt, [Instances mit Spitzenlastleistung](#).
- Verwenden Sie für zustandslose und fehlertolerante Workloads [Amazon EC2 Spot-Instances](#), um die allgemeine Nutzung der Cloud zu verbessern und die nachhaltigkeitsbezogenen Auswirkungen nicht genutzter Ressourcen zu reduzieren.
- Betreiben und optimieren Sie Ihre Workload-Instance.
  - Prüfen Sie für kurz andauernde Workloads [Amazon CloudWatch-Instance-Metriken](#) wie die CPU-Nutzung, um festzustellen, ob die Instance eventuell zu wenig oder gar nicht genutzt wird.
  - Prüfen Sie für stabile Workloads AWS-Dimensionierungstools wie etwa [AWS Compute Optimizer](#) in regelmäßigen Intervallen, um Möglichkeiten zur Optimierung und zur korrekten Dimensionierung der Instances zu erkennen.
    - [Well-Architected Lab – Empfehlungen zur Dimensionierung](#)
    - [Well-Architected Lab – Dimensionierung mit Compute Optimizer](#)
    - [Well-Architected Lab – Optimieren von Hardwaremustern und Überwachen von KPIs zur Nachhaltigkeit](#)

## Ressourcen

### Zugehörige Dokumente:

- [Optimieren Ihrer AWS-Infrastruktur für Nachhaltigkeit, Teil I: Datenverarbeitung](#)
- [AWS Graviton](#)
- [Amazon EC2 DL1](#)



- [Amazon EC2-Flotten zur Kapazitätsreservierung](#)
- [Amazon EC2-Spot-Flotte](#)
- [Funktionen: Lambda-Funktionskonfiguration](#)
- [Attribute-based instance type selection for Amazon EC2 Fleet \(Attributbasierte Auswahl des Instance-Typs für die EC2 Fleet\)](#)
- [Building Sustainable, Efficient, and Cost-Optimized Applications on AWS \(Entwicklung nachhaltiger, effizienter und kostenoptimierter Anwendungen auf AWS\)](#)
- [How the Contino Sustainability Dashboard Helps Customers Optimize Their Carbon Footprint \(So können Kunden mit dem Contino Sustainability Dashboard ihren CO2-Fußabdruck optimieren\)](#)

#### Zugehörige Videos:

- [Deep dive on AWS Graviton2 processor-powered Amazon EC2 instances \(Ein tiefer Einblick in vom Graviton2-Prozessor unterstützte Instances\)](#)
- [Deep dive into AWS Graviton3 and Amazon EC2 C7g instances \(Ein tiefer Einblick in AWS-Graviton3- und EC2-C7g-Instances\)](#)
- [Build a cost-, energy-, and resource-efficient compute environment \(Entwickeln einer kosten-, energie- und ressourceneffizienten Datenverarbeitungsumgebung\)](#)

#### Zugehörige Beispiele:

- [Lösung: Anleitung zur Optimierung von Deep-Learning-Workloads für mehr Nachhaltigkeit auf AWS](#)
- [Well-Architected Lab – Empfehlungen zur Dimensionierung](#)
- [Well-Architected Lab – Dimensionierung mit Compute Optimizer](#)
- [Well-Architected Lab – Optimieren von Hardwaremustern und Überwachen von KPIs zur Nachhaltigkeit](#)
- [Well-Architected Lab – Migration von Services zu Graviton](#)

## SUS05-BP03 Verwenden verwalteter Services

Verwenden Sie verwaltete Services für effizientere Betriebsabläufe in der Cloud.

#### Typische Anti-Muster:

- Sie verwenden Amazon EC2-Instances mit geringer Ausnutzung für die Ausführung Ihrer Anwendungen.
- Ihr internes Team verwaltet nur den Workload, ohne Zeit zu haben, sich auf Innovation oder Vereinfachungen zu konzentrieren.
- Sie nutzen und verwalten Technologien für Aufgaben, die effizienter auf verwalteten Services ausgeführt werden können.

Vorteile der Nutzung dieser bewährten Methode:

- Durch die Verwendung verwalteter Services geht die Verantwortung auf AWS über, mit Erkenntnissen zu Millionen von Kunden, was Innovationen und neue Effizienzen ermöglicht.
- Ein verwalteter Service verteilt die Umweltauswirkungen des Services durch Multi-Tenent-Steurebenen auf viele Nutzer.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: Mittel

## Implementierungsleitfaden

Verwaltete Services übertragen die Verantwortung für die Wahrung einer hohen durchschnittlichen Nutzung und die Optimierung der Nachhaltigkeit der bereitgestellten Hardware auf AWS. Verwaltete Services eliminieren dazu den betrieblichen und administrativen Aufwand für die Wartung eines Service, so Ihr Team mehr Zeit hat und sich auf Innovationen konzentrieren kann.

Prüfen Sie Ihren Workload, um die Komponenten zu identifizieren, die von verwalteten AWS-Services ersetzt werden können. Beispielsweise bieten [Amazon RDS](#), [Amazon Redshift](#) und [Amazon ElastiCache](#) einen verwalteten Datenbankservice. [Amazon Athena](#), [Amazon EMR](#) und [Amazon OpenSearch Service](#) bieten einen verwalteten Analytics-Service.

## Implementierungsschritte

1. Inventarisieren Sie Ihren Workload nach Services und Komponenten.
2. Prüfen und identifizieren Sie Komponenten, die von verwalteten Services ersetzt werden können. Hier finden Sie einige Beispiele für Situationen, in denen Sie einen verwalteten Service in Erwägung ziehen sollten:

Task	What to use on AWS
Hosten einer Datenbank	Verwenden Sie verwaltete <a href="#">Amazon Relational Database Service (Amazon RDS)</a> -Instances, anstatt Ihre eigenen Amazon RDS-Instances auf <a href="#">Amazon Elastic Compute Cloud (Amazon EC2)</a> zu verwalten.
Hosten eines Container-Workloads	Verwenden Sie <a href="#">AWS Fargate</a> , anstatt Ihre eigene Container-Infrastruktur zu implementieren.
Hosten von Web-Apps	Verwenden Sie <a href="#">AWS Amplify Hosting</a> als vollständig verwalteten CI/CD- und Hosting-Service für statische Websites und serverseitig gerenderte Web-Apps.

3. Identifizieren Sie Abhängigkeiten und erstellen Sie einen Migrationsplan. Aktualisieren Sie Runbooks und Playbooks entsprechend.
  - Der [AWS Application Discovery Service](#) erfasst und präsentiert automatisch detaillierte Informationen zu Abhängigkeiten und zur Nutzung von Anwendungen, damit Sie bei der Planung Ihrer Migration fundierte Entscheidungen treffen können.
4. Testen Sie den Service vor der Migration zum verwalteten Service.
5. Verwenden Sie den Migrationsplan zum Ersatz selbstgehosteter Services durch verwaltete Services.
6. Überwachen Sie den Service nach der Migration kontinuierlich, um erforderliche Anpassungen vorzunehmen und den Service zu optimieren.

## Ressourcen

Zugehörige Dokumente:

- [AWS Cloud-Produkte](#)
- [AWS-Gesamtbetriebskostenrechner \(Total Cost of Ownership, TCO\)](#)
- [Amazon DocumentDB](#)

- [Amazon Elastic Kubernetes Service \(EKS\)](#)
- [Amazon Managed Streaming for Apache Kafka \(Amazon MSK\)](#)

Zugehörige Videos:

- [Cloud operations at scale with AWS Managed Services](#) (Cloud-Betriebsabläufe in großem Umfang mit AWS Managed Services)

## SUS05-BP04 Optimieren der Nutzung von hardwarebasierten Computing-Beschleunigern

Sie können die Nutzung von beschleunigten Computing-Instances optimieren, um die Anforderungen Ihres Workloads an die physische Infrastruktur zu reduzieren.

Typische Anti-Muster:

- Sie überwachen die GPU-Nutzung nicht.
- Sie verwenden eine allgemeine Instance für den Workload, während eine speziell angefertigte Instance eine höhere Leistung, geringere Kosten und eine bessere Leistung pro Watt bieten kann.
- Sie verwenden hardwarebasierte Computing-beschleuniger für Aufgaben, bei denen CPU-basierte Alternativen effizienter sind.

Vorteile der Nutzung dieser bewährten Methode: Durch den optimalen Einsatz hardwarebasierter Beschleuniger können Sie die Anforderungen an die physische Infrastruktur Ihres Workloads reduzieren.

Risikostufe bei fehlender Befolgung dieser Best Practice: Mittel

### Implementierungsleitfaden

Wenn Sie eine hohe Verarbeitungsleistung benötigen, können Sie beschleunigte Computing-Instances verwenden. Diese bieten Zugriff auf hardwarebasierte Computing-Beschleuniger wie Grafikprozessoren (Graphics Processing Units, GPUs) und Field Programmable Gate Arrays (FPGAs). Diese Hardwarebeschleuniger führen bestimmte Funktionen wie die Grafikverarbeitung oder Datenmusterzuordnung effizienter aus als CPU-basierte Alternativen. Viele beschleunigte Workloads, wie Rendering, Transcodierung und Machine Learning, sind sehr variabel im Bezug auf

die Ressourcennutzung. Betreiben Sie diese Hardware nur so lange wie nötig und nehmen Sie sie automatisch außer Betrieb, wenn sie nicht mehr benötigt wird, um den Ressourcenverbrauch zu minimieren.

## Implementierungsschritte

- Identifizieren Sie, welche [beschleunigten Computing-Instances](#) Ihren Anforderungen entsprechen.
- Nutzen Sie für Machine-Learning-Workloads spezielle Hardware, die auf Ihren Workload abgestimmt ist, z. B. [AWS Trainium](#), [AWS Inferentia](#) und [Amazon EC2 DL1](#). AWS Inferentia-Instances wie Inf2-Instances bieten bis zu [50 % bessere Leistung pro Watt im Vergleich zu vergleichbaren Amazon EC2-Instances](#).
- Erfassen Sie Nutzungsmetriken für Ihre beschleunigten Computing-Instances. Sie können beispielsweise den CloudWatch-Agenten verwenden, um Metriken wie `utilization_gpu` und `utilization_memory` für Ihre GPUs zu erfassen, siehe auch [Erfassen von NVIDIA-GPU-Metriken mit Amazon CloudWatch](#).
- Optimieren Sie Code, Netzwerkbetrieb und die Einstellungen von Hardwarebeschleunigern, um sicherzustellen, dass die zugrunde liegende Hardware optimal genutzt wird.
  - [Optimieren der GPU-Einstellungen](#)
  - [GPU-Überwachung und -Optimierung im Deep-Learning-AMI](#)
  - [Optimizing I/O for GPU performance tuning of deep learning training in Amazon SageMaker \(Optimieren von E/A für die GPU-Leistungsoptimierung von Deep Learning-Training in Amazon SageMaker\)](#)
- Verwenden Sie die aktuellen leistungsstarken Bibliotheken und GPU-Treiber.
- Automatisieren Sie die Freigabe nicht genutzter GPU-Instances.

## Ressourcen

Zugehörige Dokumente:

- [Accelerated Computing](#)
- [Let's Architect! Architecting with custom chips and accelerators \(Erstellen von Architekturen mit benutzerdefinierten Chips und Beschleunigern\)](#)
- [How do I choose the appropriate Amazon EC2 instance type for my workload? \(Wie wähle ich einen geeigneten EC2-Instance-Typ für meinen Workload aus?\)](#)
- [Amazon EC2-VT1-Instances](#)

- [Choose the best AI accelerator and model compilation for computer vision inference with Amazon SageMaker \(Auswählen des besten KI-Beschleunigers und der besten Modellkompilierung für Computer Vision Inference mit Amazon SageMaker\)](#)

Zugehörige Videos:

- [How to select Amazon EC2 GPU instances for deep learning \(Auswählen von EC2-GPU-Instances für Deep Learning\)](#)
- [Deploying Cost-Effective Deep Learning Inference \(Bereitstellen von kosteneffizienten Deep Learning Inference\)](#)

## Prozess und Kultur

Reduzieren Sie nachhaltigkeitsbezogene Auswirkungen, indem Sie Ihre Entwicklungs-, Test- und Bereitstellungsmethoden ändern.

Bewährte Methoden

- [SUS06-BP01 Einführen von Methoden, die schnelle Verbesserungen für die Nachhaltigkeit ermöglichen](#)
- [SUS06-BP02 Konstantes Aktualisieren Ihres Workloads](#)
- [SUS06-BP03 Höhere Auslastung von Entwicklungsumgebungen](#)
- [SUS06-BP04 Verwenden verwalteter Gerätefarmen für Tests](#)

### SUS06-BP01 Einführen von Methoden, die schnelle Verbesserungen für die Nachhaltigkeit ermöglichen

Nutzen Sie Methoden und Prozesse zur Validierung potenzieller Verbesserung, zur Minimierung von Testkosten und zur Bereitstellung kleinerer Verbesserungen.

Typische Anti-Muster:

- Die Prüfung Ihrer Anwendung auf Nachhaltigkeitsaspekte erfolgt nur einmal zu Beginn des Projekts.
- Ihr Workload stagniert, da der Freigabeprozess zu komplex ist, um kleinere Verbesserungen für die Ressourceneffizienz umzusetzen.

- Sie verfügen über keine Mechanismen zur Verbesserung Ihres Workloads unter Nachhaltigkeitsaspekten.

Vorteile der Nutzung dieser bewährten Methode: Durch die Einrichtung eines Prozesses für die Einführung und Nachverfolgung von Nachhaltigkeitsverbesserungen können Sie kontinuierlich neue Funktionen einführen, Probleme beseitigen und die Workload-Effizienz verbessern.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: mittel

## Implementierungsleitfaden

Testen und validieren Sie potenzielle Verbesserungen in Bezug auf die Nachhaltigkeit, bevor Sie sie in der Produktion bereitstellen. Berücksichtigen Sie die Testkosten bei der Berechnung des potenziellen zukünftigen Nutzens einer Verbesserung. Entwickeln Sie kostengünstige Testmethoden, um kleinere Verbesserungen einzuführen.

### Implementierungsschritte

- Fügen Sie Ihrem Entwicklungsbacklog Anforderungen an die Nachhaltigkeit hinzu.
- Verwenden Sie einen iterativen [Verbesserungsprozess](#), um diese Verbesserungen zu identifizieren, zu bewerten, zu priorisieren, zu testen und bereitzustellen.
- Verbessern und optimieren Sie Ihre Entwicklungsprozesse kontinuierlich. Sie können beispielsweise [Ihren Softwarebereitstellungsprozess mit Pipelines für die kontinuierliche Integration und Bereitstellung \(CI/CD\)](#) automatisieren, um potenzielle Verbesserungen zu testen und bereitzustellen und so den Aufwand zu reduzieren und Fehler durch manuelle Prozesse zu minimieren.
- Testen Sie mögliche Verbesserungen mit der geringstmöglichen Zahl repräsentativer Komponenten, um die Testkosten zu reduzieren.
- Prüfen Sie kontinuierlich die Auswirkungen von Verbesserungen und nehmen Sie bei Bedarf Anpassungen vor.

## Ressourcen

Zugehörige Dokumente:

- [AWS enables sustainability solutions](#) (AWS unterstützt Lösungen für die Nachhaltigkeit)

- [Scalable agile development practices based on AWS CodeCommit](#) (Skalierbare, agile Entwicklungspraktiken auf der Grundlage von AWS CodeCommit)

Zugehörige Videos:

- [Delivering sustainable, high-performing architectures](#) (Bereitstellung nachhaltiger, leistungsstarker Architekturen)

Zugehörige Beispiele:

- [Well-Architected Lab - Turning cost & usage reports into efficiency reports](#) (Well-Architected Lab – Umwandlung von Kosten- und Nutzenberichten in Effizienzberichte)

## SUS06-BP02 Konstantes Aktualisieren Ihres Workloads

Halten Sie Ihren Workload auf neustem Stand, um effiziente Funktionen zu übernehmen, Probleme zu beseitigen und die allgemeine Effizienz des Workloads zu wahren.

Typische Anti-Muster:

- Sie gehen davon aus, dass Ihre aktuelle Architektur statisch ist und im Laufe der Zeit nicht aktualisiert wird.
- Sie haben keine Systeme oder regelmäßigen Besprechungen zur Prüfung, ob aktualisierte Software und Pakete mit Ihrem Workload kompatibel sind.

Vorteile der Einrichtung dieser bewährten Methode: Wenn Sie einen Prozess einrichten, um Ihren Workload auf neustem Stand zu halten, können Sie neue Funktionen und Kapazitäten nutzen, Probleme lösen und die Workload-Effizienz verbessern.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: niedrig

### Implementierungsleitfaden

Aktuelle Betriebssysteme, Runtimes, Middleware, Bibliotheken und Anwendungen können die Workload-Effizienz verbessern und die Nutzung effizienterer Technologien unterstützen. Aktuelle Software kann darüber hinaus Funktionen für eine genauere Messung der Auswirkungen Ihres Workloads bereitstellen, da die Anbieter mit ihrer Software ebenfalls Nachhaltigkeitsziele erfüllen



müssen. Sorgen Sie für Regelmäßigkeit bei der Aktualisierung Ihres Workloads mit den neuesten Funktionen und Versionen.

### Implementierungsschritte

- Definieren Sie einen Prozess und einen Zeitplan zur Evaluierung neuer Funktionen oder Instances für Ihre Workloads. Nutzen Sie die Agilität in der Cloud, um schnell zu testen, wie neue Funktionen Ihre Workloads auf den folgenden Gebieten verbessern können:
  - Reduzierung von Auswirkungen auf die Nachhaltigkeit.
  - Erzielen von Leistungseffizienzen.
  - Beseitigen von Hindernissen für geplante Verbesserungen.
  - Verbesserung Ihrer Fähigkeit für die Messung von und den Umgang mit Nachhaltigkeitsauswirkungen.
- Inventarisierung Ihrer Workload-Software und -Architektur und Identifizieren von Komponenten, die aktualisiert werden müssen.
  - Sie können [AWS Systems Manager Inventory](#) verwenden, um Betriebssystem (BS)-, Anwendungs- und Instance-Metadaten von Ihren Amazon EC2-Instances zu erfassen und so schnell zu erfassen, welche Instances die Software und die Konfigurationen ausführen, die Ihre Softwarerichtlinie erfordert, und welche Instances aktualisiert werden müssen.
- Verständnis der Aktualisierung der Komponenten Ihres Workloads.

Workload component	How to update
Machine Images	Verwenden Sie <a href="#">EC2 Image Builder</a> zur Verwaltung von Updates für <a href="#">Amazon Machine Images (AMIs)</a> für Linux- oder Windows Server-Images.
Container-Images	Verwenden Sie <a href="#">Amazon Elastic Container Registry (Amazon ECR)</a> mit Ihrer vorhandenen Pipeline zur <a href="#">Verwaltung Amazon Elastic Container Service (Amazon ECS) von Images</a> .
AWS Lambda	AWS Lambda enthält <a href="#">Versionsmanagement funktionen</a> .

- Verwenden Sie Automatisierung für den Aktualisierungsvorgang, um den Aufwand für die Bereitstellung neuer Funktionen zu reduzieren und Fehler zu begrenzen, die durch manuelle Prozesse verursacht werden.
- Sie können [CI/CD](#) verwenden, um AMIs, Container-Images und andere Artefakte im Zusammenhang mit Ihrer Cloud-Anwendung automatisch zu aktualisieren.
- Sie können Tools wie den [AWS Systems Manager Patch Manager](#) verwenden, um den Systemaktualisierungsprozess zu automatisieren und die Aktivitäten mit [AWS Systems Manager Maintenance Windows](#) zu planen.

## Ressourcen

Zugehörige Dokumente:

- [AWS Architecture Center](#)
- [Neuerungen bei AWS](#)
- [AWS Entwicklertools](#)

Zugehörige Beispiele:

- [Well-Architected Labs: Bestands- und Patch-Verwaltung](#)
- [Lab: AWS Systems Manager](#)

## SUS06-BP03 Höhere Auslastung von Entwicklungsumgebungen

Erhöhen Sie die Ausnutzung von Ressourcen zum Entwickeln, Testen und Erstellen Ihrer Workloads.

Typische Anti-Muster:

- Sie stellen Ihre Build-Umgebungen manuell bereit oder beenden sie in dieser Weise.
- Sie lassen Ihre Build-Umgebungen unabhängig von Test-, Build- oder Freigabeaktivitäten laufen (dazu gehört etwa der Betrieb einer Umgebung außerhalb der Arbeitszeit der Mitglieder Ihres Entwicklungsteams).
- Sie stellen übermäßig viele Ressourcen für Ihre Build-Umgebung bereit.

Vorteile der Nutzung dieser bewährten Methode: Durch die Steigerung der Ausnutzung von Build-Umgebungen können Sie die allgemeine Effizienz Ihres Cloud-Workloads verbessern, da die Ressourcen in effizienter Weise Entwicklungs-, Test- und Build-Aktivitäten zugewiesen werden.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: niedrig

## Implementierungsleitfaden

Verwenden Sie Automatisierung und „Infrastructure as Code“, um Build-Umgebungen in Betrieb zu nehmen, wenn sie gebraucht werden, und sie andernfalls zu deaktivieren. Eine typische Vorgehensweise besteht in der Planung von Verfügbarkeitszeiten, die mit den Arbeitszeiten der Entwicklungsteams übereinstimmen. Ihre Testumgebungen sollten der Produktionskonfiguration sehr stark ähneln. Suchen Sie aber nach Möglichkeiten, Instance-Typen mit Burst-Kapazität, Amazon EC2-Spot-Instances, automatisch skalierenden Datenbankservices, Containern und Serverless-Technologien zu verwenden, um die Entwicklungs- und Testkapazität an der Nutzung auszurichten. Begrenzen Sie das Datenvolumen auf die Testanforderungen. Wenn Sie Produktionsdaten für einen Test verwenden, sollten Sie nach Möglichkeiten suchen, Daten aus der Produktion gemeinsam zu nutzen, anstatt Daten hin- und herzuschieben.

## Implementierungsschritte

- Verwenden Sie „Infrastructure as Code“ zur Bereitstellung Ihrer Build-Umgebungen.
- Nutzen Sie Automatisierungen, um den Lebenszyklus Ihrer Entwicklungs- und Testumgebungen zu verwalten und die Effizienz Ihrer Build-Ressourcen zu maximieren.
- Verwenden Sie Strategien zur Maximierung der Nutzung von Entwicklungs- und Testumgebungen.
  - Verwenden Sie die geringstmögliche Zahl repräsentativer Umgebungen, um mögliche Verbesserungen zu entwickeln und zu testen.
  - Nutzen Sie nach Möglichkeit Serverless-Technologien.
  - Verwenden Sie On-Demand-Instances, um Entwicklergeräte zu ergänzen.
  - Verwenden Sie Instance-Typen mit Burst-Kapazität, Spot Instances und andere Technologien, um die Entwicklungskapazität an der Nutzung auszurichten.
  - Nutzen Sie native Cloud-Services für den sicheren Instance-Shell-Zugriff, statt Bastion-Host-Flotten bereitzustellen.
  - Skalieren Sie Ihre Build-Ressourcen automatisch je nach Build-Aktivität.

## Ressourcen

Zugehörige Dokumente:

- [AWS Systems Manager Session Manager](#)
- [Amazon EC2 Burstable performance instances](#) (Amazon EC2-Instances mit Spitzenlastleistung)
- [Was ist AWS CloudFormation?](#)
- [Was ist AWS CodeBuild?](#)
- [Instance Scheduler on AWS](#)

Zugehörige Videos:

- [Continuous Integration Best Practices](#) (Bewährte Methoden für die kontinuierliche Integration)

## SUS06-BP04 Verwenden verwalteter Gerätefarmen für Tests

Verwenden Sie verwaltete Gerätefarmen zum effektiven Testen neuer Features auf einer repräsentativen Auswahl von Hardwaregeräten.

Typische Anti-Muster:

- Sie testen Ihre Anwendung manuell und stellen sie auf einzelnen physischen Geräten bereit.
- Sie verwenden keinen App-Testservice zum Testen und zum Interagieren mit Ihren Apps (beispielsweise Android, iOS und Web-Apps) auf realen physischen Geräten.

Vorteile der Nutzung dieser bewährten Methode: Die Verwendung verwalteter Gerätefarmen zum Testen cloud-fähiger Anwendungen bringt eine Reihe von Vorteilen mit sich:

- Dazu gehören effizientere Funktionen zum Testen von Anwendungen auf einer breiten Palette von Geräten.
- Sie machen hausinterne Infrastruktur zum Testen überflüssig.
- Sie bieten unterschiedliche Gerätetypen, darunter ältere und weniger verbreitete Hardware, was unnötige Geräte-Upgrades eliminiert.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: niedrig

## Implementierungsleitfaden

Die Verwendung verwalteter Gerätefarmen kann Ihnen dabei helfen, Ihre Testprozesse für neue Funktionen auf einer repräsentativen Auswahl von Hardwaregeräten zu optimieren. Verwaltete Gerätefarmen stellen verschiedene Gerätetypen bereit, unterstützen auch ältere und weniger verbreitete Hardware und vermeiden nachhaltigkeitsbezogene Auswirkungen auf Kunden durch unnötige Geräte-Upgrades.

### Implementierungsschritte

- Definieren Sie Ihre Testanforderungen und Ihren Testplan (etwa Testtyp, Betriebssysteme und Testzeitplan).
  - Sie können [Amazon CloudWatch RUM](#) verwenden, um clientseitige Daten zu erfassen und zu analysieren und Ihren Testplan zu entwerfen.
- Wählen Sie die verwaltete Gerätefarm, die Ihre Testanforderungen unterstützen kann. Sie können beispielsweise [AWS Device Farm](#) verwenden, um die Auswirkungen Ihrer Änderungen auf eine repräsentative Auswahl von Hardwaregeräten zu testen und zu verstehen.
- Verwenden Sie kontinuierliche Integration/Bereitstellung (CI/CD) für die Planung und Durchführung Ihrer Tests.
  - [Integration von AWS Device Farm mit Ihrer CI/CD-Pipeline zur Durchführung Browser-übergreifender Selenium-Tests](#)
  - [Erstellen und Testen von iOS- und iPadOS-Apps mit AWS DevOps und mobilen Services](#)
- Prüfen Sie kontinuierlich Ihre Testergebnisse und nehmen Sie die erforderlichen Verbesserungen vor.

## Ressourcen

### Zugehörige Dokumente:

- [AWS Device Farm-Geräteliste](#)
- [Anzeige des CloudWatch RUM-Dashboards](#)

### Zugehörige Beispiele:

- [AWS Device Farm Beispiel-App für Android](#)
- [AWS Device Farm Beispiel-App für iOS](#)

- [Appium-Web-Tests für AWS Device Farm](#)

Zugehörige Videos:

- [Optimize applications through end user insights with Amazon CloudWatch RUM](#) (Optimierung von Anwendungen durch Endbenutzereinsichten mit Amazon CloudWatch RUM)

## Fazit

Immer mehr Unternehmen setzen sich Nachhaltigkeitsziele als Reaktion auf veränderte staatliche Vorschriften, mögliche Wettbewerbsvorteile und die Nachfrage von Kunden, Mitarbeitern und Investoren. CTOs, Architekten, Entwickler und Mitglieder von Betriebsteams suchen nach Möglichkeiten, wie sie direkt zu den Nachhaltigkeitszielen ihres Unternehmens beitragen können. Mithilfe dieser Designprinzipien und bewährten Methoden, die von AWS-Services unterstützt werden, können Sie fundierte Entscheidungen treffen, die ein Gleichgewicht zwischen Sicherheit, Kosten, Leistung, Zuverlässigkeit und betrieblicher Exzellenz sowie Nachhaltigkeit für Ihre AWS Cloud-Workloads herstellen. Jede Maßnahme, die Sie ergreifen, um den Ressourcenverbrauch zu verringern und die Effizienz Ihrer Workloads zu steigern, trägt zu einer Verringerung der Umweltbelastung bei und unterstützt die umfassenderen Nachhaltigkeitsziele Ihres Unternehmens.

# Mitwirkende

An diesem Dokument haben folgende Personen mitgewirkt:

- Sam Mokhtari, Senior Efficiency Lead Solutions Architect, Amazon Web Services
- Brendan Sisson, Principal Sustainability Solutions Architect, Amazon Web Services
- Margaret O'Toole, Sustainability Tech Leader, Amazon Web Services
- Steffen Grunwald, Principal Sustainability Solutions Architect, Amazon Web Services
- Ryan Eccles, Principal Engineer, Sustainability, Amazon
- Rodney Lester, Principal Architect, Amazon Web Services
- Adrian Cockcroft, VP Sustainability Architecture, Amazon Web Services
- Ian Meyers, Director of Technology, Solutions Architecture, Amazon Web Services



# Weitere Informationen

Weitere Informationen finden Sie in:

- [AWS Well-Architected](#)
- [AWS-Architekturzentrum](#)
- [Nachhaltigkeit in der Cloud](#)
- [AWS unterstützt nachhaltige Lösungen.](#)
- [The Climate Pledge](#)
- [Ziele für nachhaltige Entwicklung der Vereinten Nationen](#)
- [Greenhouse Gas Protocol](#)

# Dokumentversionen

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

Änderung	Beschreibung	Datum
<a href="#">Risikostufen aktualisiert</a>	Geringfügige Aktualisierungen der Risikostufen bei bewährten Methoden.	October 3, 2023
<a href="#">Leitfäden zu bewährten Methoden aktualisiert</a>	Bewährte Methoden wurden mit neuen Leitfäden in den folgenden Bereichen aktualisiert: <a href="#">Ausrichtung am Bedarf</a> , <a href="#">Software und Architektur</a> , <a href="#">Daten</a> und <a href="#">Hardware und Services</a> .	July 13, 2023
<a href="#">Für neues Framework aktualisiert</a>	Bewährte Methoden mit verbindlichen Anleitungen aktualisiert und neue bewährte Methoden hinzugefügt.	April 10, 2023
<a href="#">Whitepaper aktualisiert</a>	Bewährte Methoden mit neuen Implementierungsanleitungen aktualisiert.	December 15, 2022
<a href="#">Whitepaper aktualisiert</a>	Weitere bewährte Methoden und Verbesserungspläne hinzugefügt.	October 20, 2022
<a href="#">Erstveröffentlichung</a>	Säule für Nachhaltigkeit des AWS-Well-Architected-Framework veröffentlicht.	December 2, 2021

# Hinweise

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

© 2021, Amazon Web Services, Inc. bzw. Tochtergesellschaften des Unternehmens. Alle Rechte vorbehalten.

# AWS Glossary

For the latest AWS terminology, see the [AWS glossary](#) in the AWS-Glossar Reference.