



Aufbau eines skalierbaren Web-Crawling-Systems für ESG-Daten zu AWS

# AWS Präskriptive Leitlinien



---

# AWS Präskriptive Leitlinien: Aufbau eines skalierbaren Web-Crawling-Systems für ESG-Daten zu AWS

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Die Handelsmarken und Handelsaufmachung von Amazon dürfen nicht in einer Weise in Verbindung mit nicht von Amazon stammenden Produkten oder Services verwendet werden, durch die Kunden irreführt werden könnten oder Amazon in schlechtem Licht dargestellt oder diskreditiert werden könnte. Alle anderen Handelsmarken, die nicht Eigentum von Amazon sind, gehören den jeweiligen Besitzern, die möglicherweise zu Amazon gehören oder nicht, mit Amazon verbunden sind oder von Amazon gesponsert werden.

---

# Table of Contents

Einführung .....	1
Zielgruppe .....	2
Gezielte Geschäftsergebnisse .....	2
Architektur .....	4
Design und Betrieb von Webcrawlern .....	5
Batching und Datenverarbeitung .....	7
Aufbau des Systems .....	8
Einen Datensatz vorbereiten .....	8
Den Webcrawler erstellen .....	10
Erfassen und Verarbeiten der Datei robots.txt .....	10
Erfassung und Verarbeitung der Sitemap .....	12
Den Crawler entwerfen .....	14
Aufbau der Infrastruktur AWS .....	22
Bewährte Methoden .....	23
Einhaltung von Robots.txt .....	23
Begrenzung der Crawl-Rate .....	24
Transparenz durch Benutzeragenten .....	24
Effizientes Crawlen .....	24
Adaptiver Ansatz .....	24
Fehlerbehandlung .....	24
Das Crawlen erfolgt stapelweise .....	25
Sicherheit .....	25
Weitere Überlegungen .....	25
Häufig gestellte Fragen .....	27
Was ist, wenn die Datei robots.txt nicht verfügbar ist? .....	27
Was ist, wenn eine Datei sitemaps.xml nicht verfügbar ist? .....	27
Kann ich anstelle von Amazon EC2 oder Amazon ECS eine serverlose Lösung verwenden? .....	28
Warum erhält der Crawler einen 403-Statuscode? .....	28
Nächste Schritte und Ressourcen .....	29
Ressourcen .....	29
Tools .....	30
Dokumentverlauf .....	31
Glossar .....	32
# .....	32

---

A .....	33
B .....	36
C .....	38
D .....	42
E .....	46
F .....	48
G .....	50
H .....	51
I .....	53
L .....	56
M .....	57
O .....	61
P .....	64
Q .....	67
R .....	68
S .....	71
T .....	75
U .....	77
V .....	77
W .....	78
Z .....	79
.....	lxxx

# Aufbau eines skalierbaren Web-Crawling-Systems für ESG-Daten zu AWS

Vijit Vashishtha und Mansi Doshi, Amazon Web Services

[Januar 2025 \(Geschichte der Dokumente\)](#)

Umwelt-, Sozial- und Unternehmensführungsfaktoren (ESG) sind wichtige Überlegungen für Anleger bei der Bewertung potenzieller Anlagen:

- Umwelt — Konzentriert sich auf die Auswirkungen eines Unternehmens auf die Natur. Es umfasst Faktoren wie CO2-Emissionen, Ressourcenmanagement und Energieeffizienz.
- Soziales — Untersucht, wie ein Unternehmen die Beziehungen zu Mitarbeitern, Lieferanten, Kunden und Gemeinschaften verwaltet. Es deckt Aspekte wie Arbeitspraktiken, Vielfalt und gesellschaftliches Engagement ab.
- Unternehmensführung — Betrachtet die Führung, die internen Kontrollen und die Aktionärsrechte eines Unternehmens. Dazu gehören die Zusammensetzung des Verwaltungsrats, die Vergütung von Führungskräften und die Geschäftsethik.

Unternehmen mit soliden ESG-Praktiken werden zunehmend als besser positioniert angesehen, wenn es um langfristige Nachhaltigkeit und Rentabilität geht. Die Nachfrage der Anleger nach ESG-Informationen wächst. Unternehmen, die ihre Nachhaltigkeitsbilanz anhand zuverlässiger, nützlicher ESG-Daten nachweisen können, sind besser positioniert, um Kapital anzuziehen und wettbewerbsfähig zu bleiben. Unternehmen veröffentlichen ESG-Daten über verschiedene Quellen wie Nachrichten, Artikel und Jahresberichte. Da diese Informationen verstreut sind, kann Ihnen ein Webcrawler dabei helfen, diese Daten effizient zu sammeln.

Dieser umfassende Leitfaden zeigt [AWS Fargate](#), wie Sie [Amazon Elastic Compute Cloud \(Amazon EC2\)](#) und Amazon [Simple Storage Service \(Amazon S3\)](#) verwenden [AWS Batch](#), um eine robuste, skalierbare und verantwortungsvolle Datenerfassungspipeline aufzubauen. Es behandelt Folgendes:

- Aufbau eines skalierbaren Crawling-Systems mithilfe der folgenden Komponenten: AWS-Services
  - Fargate oder Amazon EC2 zum Ausführen der Crawler-Anwendung
  - AWS Batch für die effiziente Orchestrierung umfangreicher Crawling-Jobs
  - Amazon S3 für sichere und dauerhafte Datenspeicherung

- Implementierung von Best Practices für ethisches Crawling, darunter:
  - Einhaltung der Richtlinien von robots.txt und der Website
  - Verwaltung der Ratenbegrenzung, um eine Überlastung der Zielseiten zu vermeiden
  - Gewährleistung des Datenschutzes und verantwortungsvoller Umgang mit den gesammelten Informationen
- Entwicklung eines Python basierten Crawlers, der für AWS die Infrastruktur optimiert ist
- Optimierung der Crawler-Leistung unter Einhaltung ethischer Standards

## Zielgruppe

Dieser Leitfaden richtet sich an Dateningenieure und Cloud-Architekten, die effizient große Mengen an up-to-date ESG-Daten von öffentlichen Websites sammeln möchten. Er ist besonders relevant für Projekte, die Marktanalysen, nachhaltige Finanzanalysen oder Finanzanalysen beinhalten.

## Gezielte Geschäftsergebnisse

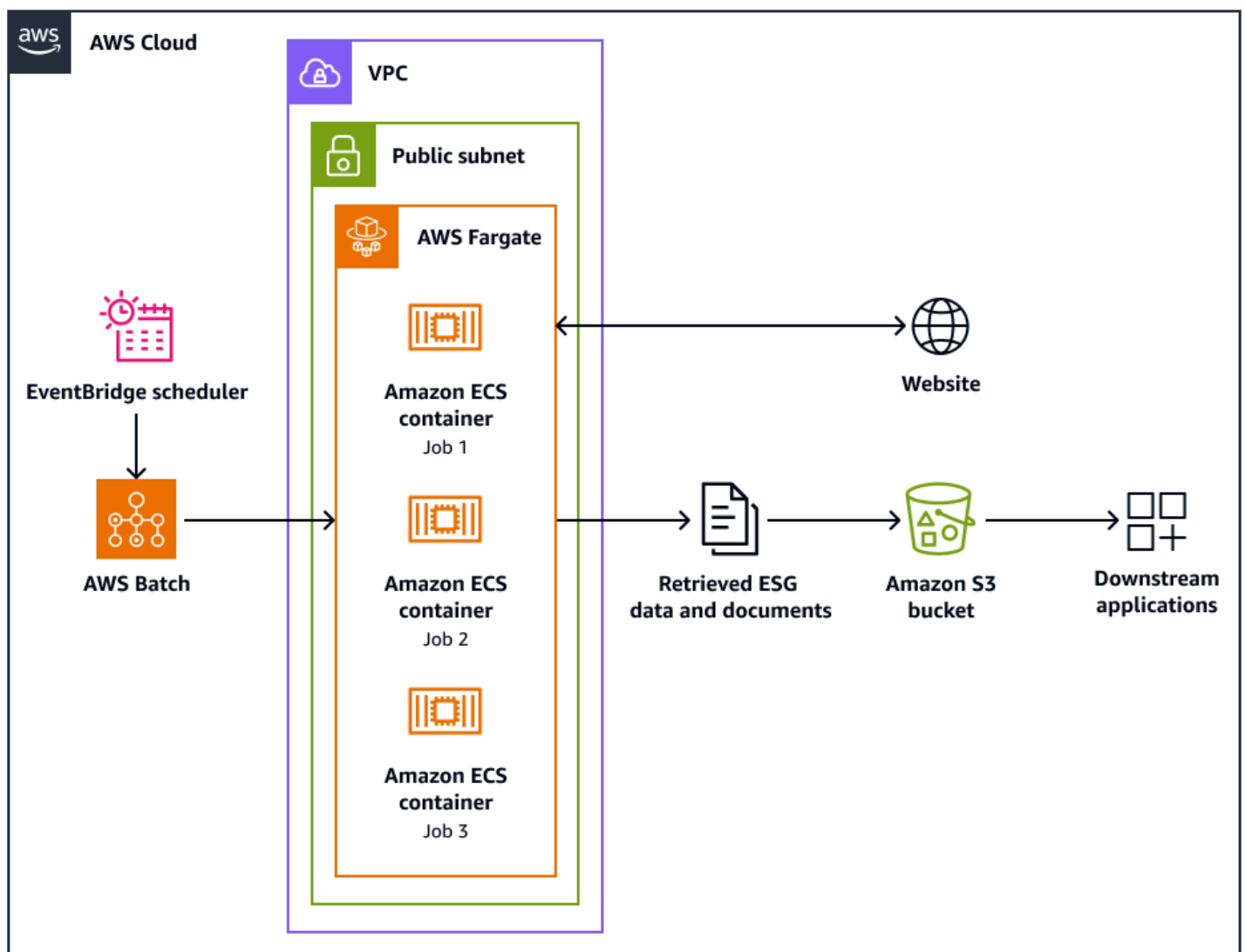
Die folgenden Gründe sprechen häufig dafür, dass Unternehmen ESG-Daten verwenden:

- Risikomanagement — ESG-Daten helfen Ihnen dabei, potenzielle Risiken im Zusammenhang mit Umwelt-, Sozial- und Unternehmensführungsfragen zu identifizieren und zu mindern.
- Anziehung von Anlegern — Viele Anleger berücksichtigen heute ESG-Faktoren, wenn sie Anlageentscheidungen treffen. Sie betrachten starke ESG-Praktiken als Indikatoren für langfristige Nachhaltigkeit und Rentabilität.
- Reputationsmanagement — Eine gute ESG-Leistung kann den Ruf eines Unternehmens bei Kunden, Mitarbeitern und der Öffentlichkeit verbessern.
- Einhaltung gesetzlicher Vorschriften — Angesichts der zunehmenden ESG-Vorschriften hilft die Einführung von ESG-Praktiken Unternehmen, den Compliance-Anforderungen immer einen Schritt voraus zu sein.
- Innovation und Effizienz — Die Konzentration auf ESG-Faktoren kann Innovationen bei Produkten, Dienstleistungen und Abläufen vorantreiben. Dies führt zu verbesserter Effizienz und Kosteneinsparungen.
- Wettbewerbsvorteil — Eine starke ESG-Leistung kann ein Unternehmen von seinen Mitbewerbern abheben und neue Marktchancen eröffnen.

- 
- Einbindung von Stakeholdern — ESG-Praktiken helfen Unternehmen dabei, besser mit verschiedenen Interessengruppen wie Mitarbeitern, Kunden und lokalen Gemeinschaften in Kontakt zu treten und deren Erwartungen zu erfüllen.

# Architektur für ein skalierbares Web-Crawling-System auf AWS

Das folgende Architekturdiagramm zeigt ein Webcrawler-System, das darauf ausgelegt ist, Umwelt-, Sozial- und Unternehmensführungsdaten (ESG) auf ethische Weise von Websites zu extrahieren. Sie verwenden einen Python basierten Crawler, der für die Infrastruktur optimiert ist. AWS Sie AWS Batch orchestrieren die groß angelegten Crawling-Jobs und verwenden Amazon Simple Storage Service (Amazon S3) für die Speicherung. Downstream-Anwendungen können die Daten aus dem Amazon S3 S3-Bucket aufnehmen und speichern.



Das Diagramm zeigt den folgenden Workflow:

1. Amazon EventBridge Scheduler initiiert den Crawling-Prozess in einem von Ihnen geplanten Intervall.
2. AWS Batch verwaltet die Ausführung der Webcrawler-Jobs. Die AWS Batch Job-Warteschlange enthält und orchestriert die ausstehenden Crawling-Jobs.
3. Die Web-Crawling-Jobs werden in Amazon Elastic Container Service (Amazon ECS) -Containern auf AWS Fargate ausgeführt. Die Jobs werden in einem öffentlichen Subnetz einer Virtual Private Cloud (VPC) ausgeführt.
4. Der Webcrawler durchsucht die Zielwebsite und ruft ESG-Daten und Dokumente wie PDF-, CSV- oder andere Dokumentdateien ab.
5. Der Webcrawler speichert die abgerufenen Daten und Rohdateien in einem Amazon S3 S3-Bucket.
6. Andere Systeme oder Anwendungen erfassen oder verarbeiten die gespeicherten Daten und Dateien im Amazon S3 S3-Bucket.

## Design und Betrieb von Webcrawlern

Einige Websites wurden speziell für die Ausführung auf Desktops oder Mobilgeräten entwickelt. Der Webcrawler ist so konzipiert, dass er die Verwendung eines Desktop-Benutzeragenten oder eines mobilen Benutzeragenten unterstützt. Diese Agenten helfen Ihnen dabei, erfolgreich Anfragen an die Zielwebsite zu stellen.

Nachdem der Webcrawler initialisiert wurde, führt er die folgenden Operationen aus:


1. Der Webcrawler ruft die Methode auf. `setup()` Diese Methode ruft die Datei `robots.txt` ab und analysiert sie.

### Note

Sie können den Webcrawler auch so konfigurieren, dass er die Sitemap abrufen und analysiert.

2. Der Webcrawler verarbeitet die Datei `robots.txt`. Wenn in der Datei `robots.txt` eine Crawlverzögerung angegeben ist, extrahiert der Webcrawler die Crawlverzögerung für den Desktop-Benutzeragenten. Wenn in der Datei `robots.txt` keine Crawlverzögerung angegeben ist, verwendet der Webcrawler eine zufällige Verzögerung.

3. Der Webcrawler ruft die `crawl()` Methode auf, die den Crawling-Prozess initiiert. Wenn URLs sich keine in der Warteschlange befinden, wird die Start-URL hinzugefügt.

 Note

Der Crawler fährt fort, bis er die maximale Anzahl an Seiten erreicht hat oder die Crawling-Zeit URLs knapp wird.

4. Der Crawler verarbeitet die URLs Für jede URL in der Warteschlange prüft der Crawler, ob die URL bereits gecrawlt wurde.
5. Wenn eine URL nicht gecrawlt wurde, ruft der Crawler die Methode wie folgt auf `crawl_url()`:
  - a. Der Crawler überprüft die Datei `robots.txt`, um festzustellen, ob er den Desktop-Benutzeragenten zum Crawlen der URL verwenden kann.
  - b. Falls zulässig, versucht der Crawler, die URL mithilfe des Desktop-Benutzeragenten zu crawlen.
  - c. Wenn dies nicht erlaubt ist oder wenn der Desktop-Benutzeragent das Crawlen fehlschlägt, überprüft der Crawler die Datei `robots.txt`, um festzustellen, ob er den mobilen Benutzeragenten zum Crawlen der URL verwenden kann.
  - d. Falls zulässig, versucht der Crawler, die URL mithilfe des Mobile User Agents zu crawlen.
6. Der Crawler ruft die `attempt_crawl()` Methode auf, die den Inhalt abrufen und verarbeitet. Der Crawler sendet eine GET-Anfrage mit entsprechenden Headern an die URL. Wenn die Anfrage fehlschlägt, verwendet der Crawler die Wiederholungslogik.
7. Wenn die Datei im HTML-Format vorliegt, ruft der Crawler die Methode auf `extract_esg_data()` Es wird verwendet [Beautiful Soup](#), um den HTML-Inhalt zu analysieren. Es extrahiert Umwelt-, Sozial- und Unternehmensführungsdaten (ESG) mithilfe von Schlüsselwörtern.

Wenn es sich bei der Datei um eine PDF-Datei handelt, ruft der Crawler die Methode `save_pdf()`. Der Crawler lädt die PDF-Datei herunter und speichert sie im Amazon S3 S3-Bucket.

8. Der Crawler ruft die Methode auf `extract_news_links()`. Dadurch werden Links zu Nachrichtenartikeln, Pressemitteilungen und Blogbeiträgen gefunden und gespeichert.
9. Der Crawler ruft die `extract_pdf_links()` Methode auf. Dadurch werden Links zu PDF-Dokumenten identifiziert und gespeichert.

- 10 Der Crawler ruft die `is_relevant_to_sustainable_finance()` Methode auf. Dabei wird anhand vordefinierter Stichwörter geprüft, ob sich die Nachrichten oder Artikel auf nachhaltiges Finanzwesen beziehen.
- 11 Nach jedem Crawlversuch implementiert der Crawler mithilfe der `delay()` Methode eine Verzögerung. Wenn in der Datei `robots.txt` eine Verzögerung angegeben wurde, wird dieser Wert verwendet. Andernfalls wird eine zufällige Verzögerung zwischen 1 und 3 Sekunden verwendet.
- 12 Der Crawler ruft die `save_esg_data()` Methode auf, um die ESG-Daten in einer CSV-Datei zu speichern. Die CSV-Datei wird im Amazon S3 S3-Bucket gespeichert.
- 13 Der Crawler ruft die `save_news_links()` Methode auf, um die News-Links einschließlich Relevanzinformationen in einer CSV-Datei zu speichern. Die CSV-Datei wird im Amazon S3 S3-Bucket gespeichert.
- 14 Der Crawler ruft die `save_pdf_links()` Methode zum Speichern der PDF-Links in einer CSV-Datei auf. Die CSV-Datei wird im Amazon S3 S3-Bucket gespeichert.

## Batching und Datenverarbeitung

Der Crawling-Prozess wird strukturiert organisiert und durchgeführt. AWS Batch weist die Jobs für jedes Unternehmen so zu, dass sie parallel und stapelweise ausgeführt werden. Jeder Batch konzentriert sich auf die Domain und die Subdomänen eines einzelnen Unternehmens, so wie Sie sie in Ihrem Datensatz identifiziert haben. Jobs im selben Batch werden jedoch sequenziell ausgeführt, sodass die Website nicht mit zu vielen Anfragen überflutet wird. Dies hilft der Anwendung, den Crawling-Workload effizienter zu verwalten und sicherzustellen, dass alle relevanten Daten für jedes Unternehmen erfasst werden.

Durch die Organisation des Webcrawlings in unternehmensspezifische Batches werden die gesammelten Daten containerisiert. Dadurch wird verhindert, dass die Daten eines Unternehmens mit Daten anderer Unternehmen vermischt werden.

Durch Batching kann die Anwendung Daten effizient aus dem Internet sammeln und gleichzeitig eine klare Struktur und Trennung der Informationen auf der Grundlage der Zielunternehmen und ihrer jeweiligen Webdomänen beibehalten. Dieser Ansatz trägt dazu bei, die Integrität und Nutzbarkeit der gesammelten Daten zu gewährleisten, da sie übersichtlich organisiert und den entsprechenden Unternehmen und Domänen zugeordnet sind.

# Aufbau eines skalierbaren Web-Crawling-Systems auf AWS

In diesem Abschnitt wird beschrieben, wie der im Abschnitt beschriebene Webcrawler erstellt wird. [Architektur](#) Es beinhaltet einen systematischen Ansatz zur Erstellung eines soliden Datensatzes von Unternehmen und den zugehörigen Webeigenschaften. Dieser Datensatz dient als Grundlage für Ihre Crawling-Aktivitäten. Anschließend wird in diesem Abschnitt beschrieben, wie Sie einen ethischen Webcrawler einrichten. Python

## Themen

- [Einen Datensatz vorbereiten](#)
- [Den Webcrawler erstellen](#)
- [Aufbau der Infrastruktur AWS](#)

## Einen Datensatz vorbereiten

Falls Sie dies noch nicht getan haben, bereiten Sie einen detaillierten Datensatz der Websites vor, von denen Sie Informationen sammeln möchten. Dieser Datensatz sollte URL-Domainnamen der Websites und relevante Subdomainnamen enthalten. In diesem Abschnitt wird ein step-by-step Verfahren zum Erstellen dieses Datensatzes beschrieben.

Um einen Datensatz vorzubereiten

1. Definieren Sie den Umfang — Bestimmen Sie die Branche oder Sektoren, auf die Sie sich konzentrieren. Entscheiden Sie, wie viele Unternehmen Sie einbeziehen möchten. Und definieren Sie alle Kriterien, die Sie über diese Unternehmen erheben möchten, z. B. die Anzahl der Mitarbeiter, den Standort oder den Umsatz.
2. Identifizieren Sie Datenquellen — Identifizieren Sie, welche Informationsquellen Sie verwenden können, um Informationen über diese Unternehmen zu sammeln. Beispiele hierfür sind Unternehmensverzeichnisse (wie [Crunchbase](#), [Bloomberg](#) oder [Forbes](#)), Börsen (wie NYSE und NASDAQ), branchenspezifische Verbände oder Veröffentlichungen oder Regierungsdatenbanken (wie SEC-Unterlagen).
3. Erstellen Sie eine Tabelle — Erstellen Sie in Ihrem bevorzugten Tool wie Microsoft Excel, Google Sheets oder einem Datenbankverwaltungssystem eine Tabelle zum Sammeln von Kriterien für jedes Unternehmen. Fügen Sie für jedes Kriterium eine Spalte hinzu. Fügen Sie mindestens

Spalten für den Firmennamen, die Hauptdomain, die Subdomänen, die Branche, die Größe und den Standort hinzu.

4. Erste Unternehmensinformationen sammeln — Sammeln Sie die folgenden Informationen zu jedem Unternehmen und geben Sie sie in die von Ihnen erstellte Tabelle ein:
  - Unternehmensname
  - Branche oder Sektor
  - Unternehmensgröße (Anzahl der Mitarbeiter)
  - Revenue (Umsatz)
  - Standort des Firmensitzes
5. Domaininformationen sammeln — Extrahieren Sie für jedes Unternehmen den primären Domainnamen aus der URL der Haupt-Website, z. `example.com`. Sie können die Domaininformationen mithilfe eines WHOIS-Domain-Suchtools überprüfen.
6. Sammeln Sie Subdomain-Informationen — Recherchieren Sie für jedes Unternehmen nach den registrierten Subdomains, wie z. `blog.example.com` [Sie können Tools zur Subdomain-Aufzählung wie Sublist3R, OWASP Amass oder Subfinder verwenden](#). Sie können Google-Dorking (durch `Suchensite:example.com`) durchführen, DNS-Einträge mithilfe eines `dig` Befehls oder eines DNS-Suchtools überprüfen oder SSL- oder TLS-Zertifikate analysieren.
7. Daten validieren und bereinigen — Überprüfe, verifiziere und standardisiere die Daten, die du gesammelt hast. Entfernen Sie beispielsweise alle doppelten Einträge, entfernen Sie unnötige URL-Informationen aus Domains und Subdomains und stellen Sie sicher, dass alle Domains und Subdomains aktiv sind.
8. (Optional) Kategorisieren Sie die Subdomains — Sie können die Subdomains in Typen einteilen. Im Folgenden finden Sie einige Beispiele für Kategorien, auf die Sie stoßen könnten:
  - Blogs, wie `blog.example.com`
  - Support oder Hilfe, wie zum Beispiel `support.example.com` oder `help.example.com`
  - E-Commerce, wie `shop.example.com` oder `store.example.com`
  - Ressourcen für Entwickler, wie `dev.example.com` oder `api.example.com`
  - Regionen oder Standorte, wie `us.example.com` oder `uk.example.com`
9. (Optional) Fügen Sie relevante Metadaten hinzu — Sie können alle relevanten Metadaten im Datensatz aufzeichnen. Sie können beispielsweise das Datum der letzten Aktualisierung, die Informationsquelle oder Ihren Vertrauenswert für die Genauigkeit der Subdomain hinzufügen.

10. Versionskontrolle implementieren — Verwenden Sie ein Versionskontrollsystem wie Git, um Änderungen an der Tabelle im Laufe der Zeit nachzuverfolgen. Erstellen Sie regelmäßig eine Sicherungskopie des Datensatzes.
11. Tabelle verwalten — Richten Sie einen Zeitplan für die Aktualisierung der Tabelle ein, z. B. vierteljährlich. Standardisieren und implementieren Sie einen Prozess zum Hinzufügen neuer Unternehmen oder zum Entfernen von Unternehmen, die Sie nicht mehr benötigen. Automatisieren Sie nach Möglichkeit die Erkennung von Subdomains.

## Den Webcrawler erstellen

Wie im [Architektur](#) Abschnitt beschrieben, wird die Anwendung in Batches ausgeführt — einer für jedes Unternehmen.

Themen

- [Erfassen und Verarbeiten der Datei robots.txt](#)
- [Erfassung und Verarbeitung der Sitemap](#)
- [Den Crawler entwerfen](#)

## Erfassen und Verarbeiten der Datei robots.txt

Nachdem Sie den Datensatz vorbereitet haben, müssen Sie überprüfen, ob die Domain über eine Datei [robots.txt](#) verfügt. Bei Webcrawlern und anderen Bots gibt die Datei robots.txt an, welche Bereiche der Website sie besuchen dürfen. Die Einhaltung der Anweisungen in dieser Datei ist eine wichtige bewährte Methode für das ethische Crawlen von Websites. Weitere Informationen finden Sie in diesem Leitfaden unter [Bewährte Methoden für ethische Webcrawler](#).

Um die Datei robots.txt zu erfassen und zu verarbeiten

1. Falls Sie dies noch nicht getan haben, installieren Sie die `requests` Bibliothek, indem Sie den folgenden Befehl in einem Terminal ausführen:

```
pip install requests
```

2. Führen Sie das folgende Skript aus. Dieses Skript führt folgende Aktionen aus:
  - Es definiert eine `check_robots_txt` Funktion, die eine Domain als Eingabe verwendet.

- Es erstellt die vollständige URL für die Datei robots.txt.
- Es sendet eine GET-Anfrage an die URL für die Datei robots.txt.
- Wenn die Anfrage erfolgreich ist (Statuscode 200), ist eine Datei robots.txt vorhanden.
- Wenn die Anfrage fehlschlägt oder einen anderen Statuscode zurückgibt, ist die Datei robots.txt nicht vorhanden oder es kann nicht darauf zugegriffen werden.

```
import requests
from urllib.parse import urljoin
def check_robots_txt(domain):
    # Ensure the domain starts with a protocol
    if not domain.startswith(('http://', 'https://')):
        domain = 'https://' + domain
    # Construct the full URL for robots.txt
    robots_url = urljoin(domain, '/robots.txt')
    try:
        # Send a GET request to the robots.txt URL
        response = requests.get(robots_url, timeout=5)
        # Check if the request was successful (status code 200)
        if response.status_code == 200:
            print(f"robots.txt found at {robots_url}")
            return True
        else:
            print(f"No robots.txt found at {robots_url} (Status code:
{response.status_code})")
            return False
    except requests.RequestException as e:
        print(f"Error checking {robots_url}: {e}")
        return False
```

#### Note

Dieses Skript behandelt Ausnahmen für Netzwerkfehler oder andere Probleme.

3. Wenn eine Datei robots.txt existiert, verwenden Sie das folgende Skript, um sie herunterzuladen:

```
import requests

def download(self, url):
    response = requests.get(url, headers=self.headers, timeout=5)
```

```
response.raise_for_status() # Raise an exception for non-2xx responses
return response.text

def download_robots_txt(self):
    # Append '/robots.txt' to the URL to get the robots.txt file's URL
    robots_url = self.url.rstrip('/') + '/robots.txt'
    try:
        response = download(robots_url)
        return response
    except requests.exceptions.RequestException as e:
        print(f"Error downloading robots.txt: {e}, \nGenerating sitemap using
combinations...")
        return e
```

### Note

Diese Skripts können je nach Anwendungsfall angepasst oder geändert werden. Sie können diese Skripte auch kombinieren.

## Erfassung und Verarbeitung der Sitemap

Als Nächstes müssen Sie die [Sitemap](#) bearbeiten. Sie können die Sitemap verwenden, um das Crawlen auf wichtige Seiten zu konzentrieren. Dies verbessert die Crawling-Effizienz. Weitere Informationen finden Sie in diesem Leitfaden unter [Bewährte Methoden für ethische Webcrawler](#).

Um die Sitemap zu erfassen und zu verarbeiten

- Führen Sie das folgende Skript aus. Dieses Skript definiert eine `check_and_download_sitemap` Funktion, die:
  - Akzeptiert eine Basis-URL, eine optionale Sitemap-URL aus robots.txt und eine User-Agent-Zeichenfolge.
  - Überprüft mehrere potenzielle Sitemap-Speicherorte, einschließlich des Speicherorts von robots.txt (falls angegeben).
  - Versucht, die Sitemap von jedem Standort herunterzuladen.
  - Überprüft, ob der heruntergeladene Inhalt im XML-Format vorliegt.
  - Ruft die `parse_sitemap` Funktion zum Extrahieren von auf URLs. Diese Funktion:

- Analysiert den XML-Inhalt der Sitemap.
- Verarbeitet sowohl reguläre Sitemaps als auch Sitemap-Indexdateien.
- Ruft rekursiv Unter-Sitemaps ab, wenn ein Sitemap-Index gefunden wird.

```
import requests
from urllib.parse import urljoin
import xml.etree.ElementTree as ET

def check_and_download_sitemap(base_url, robots_sitemap_url=None,
    user_agent='SitemapBot/1.0'):
    headers = {'User-Agent': user_agent}
    sitemap_locations = [robots_sitemap_url, urljoin(base_url, '/sitemap.xml'),
    urljoin(base_url, '/sitemap_index.xml'),
    urljoin(base_url, '/sitemap/'), urljoin(base_url, '/sitemap/sitemap.xml')]

    for sitemap_url in sitemap_locations:
        if not sitemap_url:
            continue
        print(f"Checking for sitemap at: {sitemap_url}")
        try:
            response = requests.get(sitemap_url, headers=headers, timeout=10)
            if response.status_code == 200:
                content_type = response.headers.get('Content-Type', '')
                if 'xml' in content_type:
                    print(f"Successfully downloaded sitemap from {sitemap_url}")
                    return parse_sitemap(response.text)
                else:
                    print(f"Found content at {sitemap_url}, but it's not XML.
Content-Type: {content_type}")
            except requests.RequestException as e:
                print(f"Error downloading sitemap from {sitemap_url}: {e}")
        print("No sitemap found.")
        return []

def parse_sitemap(sitemap_content):
    urls = []
    try:
        root = ET.fromstring(sitemap_content)
        # Handle both sitemap and sitemapindex
        for loc in root.findall('.://{http://www.sitemaps.org/schemas/
sitemap/0.9}loc'):
```

```
        urls.append(loc.text)

    # If it's a sitemap index, recursively fetch each sitemap
    if root.tag.endswith('sitemapindex'):
        all_urls = []
        for url in urls:
            print(f"Fetching sub-sitemap: {url}")
            sub_sitemap_urls = check_and_download_sitemap(url)
            all_urls.extend(sub_sitemap_urls)
        return all_urls
    except ET.ParseError as e:
        print(f"Error parsing sitemap XML: {e}")
    return urls

if __name__ == "__main__":
    base_url = input("Enter the base URL of the website: ")
    robots_sitemap_url = input("Enter the sitemap URL from robots.txt (or press
Enter if none): ").strip() or None
    urls = check_and_download_sitemap(base_url, robots_sitemap_url)
    print(f"Found {len(urls)} URLs in sitemap:")
    for url in urls[:5]: # Print first 5 URLs as an example
        print(url)
    if len(urls) > 5:
        print("...")
```

## Den Crawler entwerfen

Als Nächstes entwerfen Sie den Webcrawler. Der Crawler ist so konzipiert, dass er den [Bewährte Methoden für ethische Webcrawler](#) in diesem Handbuch beschriebenen bewährten Methoden folgt. In diesem EthicalCrawler Kurs werden mehrere wichtige Prinzipien des ethischen Crawlings vorgestellt:

- Die Datei robots.txt abrufen und analysieren — Der Crawler ruft die Datei robots.txt für die Zielwebsite ab.
- Einhaltung der Crawling-Berechtigungen — Vor dem Crawlen einer URL überprüft der Crawler, ob die Regeln in der Datei robots.txt das Crawlen für diese URL zulassen. Wenn eine URL nicht erlaubt ist, überspringt der Crawler sie und wechselt zur nächsten URL.

- Berücksichtigung der Crawlverzögerung — Der Crawler sucht in der Datei robots.txt nach einer Crawl-Delay-Direktive. Wenn eine angegeben ist, verwendet der Crawler diese Verzögerung zwischen Anfragen. Andernfalls verwendet er eine Standardverzögerung.
- User-Agent-Identifikation — Der Crawler verwendet eine benutzerdefinierte User-Agent-Zeichenfolge, um sich gegenüber Websites zu identifizieren. Bei Bedarf können Webseitenbesitzer spezifische Regeln festlegen, um deinen Crawler einzuschränken oder zuzulassen.
- Fehlerbehandlung und ordentliche Degradierung — Wenn die Datei robots.txt nicht abgerufen oder analysiert werden kann, verwendet der Crawler konservative Standardregeln. Er behandelt Netzwerkfehler und HTTP-Antworten, die nicht 200 sind.
- Eingeschränktes Crawlen — Um eine Überlastung des Servers zu vermeiden, gibt es ein Limit für die Anzahl der Seiten, die gecrawlt werden können.

Das folgende Skript ist Pseudocode, der erklärt, wie der Webcrawler funktioniert:

```
import requests
from urllib.parse import urljoin, urlparse
import time

class EthicalCrawler:
    def __init__(self, start_url, user_agent='EthicalBot/1.0'):
        self.start_url = start_url
        self.user_agent = user_agent
        self.domain = urlparse(start_url).netloc
        self.robots_parser = None
        self.crawl_delay = 1 # Default delay in seconds

    def can_fetch(self, url):
        if self.robots_parser:
            return self.robots_parser.allowed(url, self.user_agent)
        return True # If no robots.txt, assume allowed but crawl conservatively

    def get_crawl_delay(self):
        if self.robots_parser:
            delay = self.robots_parser.agent(self.user_agent).delay
            if delay is not None:
                self.crawl_delay = delay
        print(f"Using crawl delay of {self.crawl_delay} seconds")

    def crawl(self, max_pages=10):
        self.get_crawl_delay()
```

```
pages_crawled = 0
urls_to_crawl = [self.start_url]
while urls_to_crawl and pages_crawled < max_pages:
    url = urls_to_crawl.pop(0)
    if not self.can_fetch(url):
        print(f"robots.txt disallows crawling: {url}")
        continue
    try:
        response = requests.get(url, headers={'User-Agent': self.user_agent})
        if response.status_code == 200:
            print(f"Successfully crawled: {url}")
            # Here you would typically parse the content, extract links, etc.
            # For this example, we'll just increment the counter
            pages_crawled += 1
        else:
            print(f"Failed to crawl {url}: HTTP {response.status_code}")
    except Exception as e:
        print(f"Error crawling {url}: {e}")

    # Respect the crawl delay
    time.sleep(self.crawl_delay)

print(f"Crawling complete. Crawled {pages_crawled} pages.")
```

Um einen fortschrittlichen, ethischen Webcrawler zu entwickeln, der ESG-Daten sammelt

1. Kopieren Sie das folgende Codebeispiel für den fortschrittlichen ethischen Webcrawler, der in diesem System verwendet wird:

```
import requests
from urllib.parse import urljoin, urlparse
import time
from collections import deque
import random
from bs4 import BeautifulSoup
import re
import csv
import os

class EnhancedESGCrawler:
    def __init__(self, start_url):
```

```
self.start_url = start_url
self.domain = urlparse(start_url).netloc
self.desktop_user_agent = 'ESGEthicalBot/1.0'
self.mobile_user_agent = 'Mozilla/5.0 (iPhone; CPU iPhone OS 14_0 like
Mac OS X) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/14.0 Mobile/15E148
Safari/604.1'
self.robots_parser = None
self.crawl_delay = None
self.urls_to_crawl = deque()
self.crawled_urls = set()
self.max_retries = 2
self.session = requests.Session()
self.esg_data = []
self.news_links = []
self.pdf_links = []

def setup(self):
    self.fetch_robots_txt() # Provided in Previous Snippet
    self.fetch_sitemap() # Provided in Previous Snippet

def can_fetch(self, url, user_agent):
    if self.robots_parser:
        return self.robots_parser.allowed(url, user_agent)
    return True

def delay(self):
    if self.crawl_delay is not None:
        time.sleep(self.crawl_delay)
    else:
        time.sleep(random.uniform(1, 3))

def get_headers(self, user_agent):
    return {'User-Agent': user_agent,
            'Accept': 'text/html,application/xhtml+xml,application/
xml;q=0.9,image/webp,*/*;q=0.8',
            'Accept-Language': 'en-US,en;q=0.5', 'Accept-Encoding': 'gzip,
deflate, br', 'DNT': '1',
            'Connection': 'keep-alive', 'Upgrade-Insecure-Requests': '1'}

def extract_esg_data(self, url, html_content):
    soup = BeautifulSoup(html_content, 'html.parser')
    esg_data = {
        'url': url,
```

```
        'environmental': self.extract_environmental_data(soup),
        'social': self.extract_social_data(soup),
        'governance': self.extract_governance_data(soup)
    }
    self.esg_data.append(esg_data)
    # Extract news links and PDFs
    self.extract_news_links(soup, url)
    self.extract_pdf_links(soup, url)

    def extract_environmental_data(self, soup):
        keywords = ['carbon footprint', 'emissions', 'renewable energy', 'waste
management', 'climate change']
        return self.extract_keyword_data(soup, keywords)

    def extract_social_data(self, soup):
        keywords = ['diversity', 'inclusion', 'human rights', 'labor practices',
'community engagement']
        return self.extract_keyword_data(soup, keywords)

    def extract_governance_data(self, soup):
        keywords = ['board structure', 'executive compensation', 'shareholder
rights', 'ethics', 'transparency']
        return self.extract_keyword_data(soup, keywords)

    def extract_keyword_data(self, soup, keywords):
        text = soup.get_text().lower()
        return {keyword: len(re.findall(r'\b' + re.escape(keyword) + r'\b', text))
for keyword in keywords}

    def extract_news_links(self, soup, base_url):
        news_keywords = ['news', 'press release', 'article', 'blog',
'sustainability']
        for a in soup.find_all('a', href=True):
            if any(keyword in a.text.lower() for keyword in news_keywords):
                full_url = urljoin(base_url, a['href'])
                if full_url not in self.news_links:
                    self.news_links.append({'url': full_url, 'text':
a.text.strip()})

    def extract_pdf_links(self, soup, base_url):
        for a in soup.find_all('a', href=True):
            if a['href'].lower().endswith('.pdf'):
                full_url = urljoin(base_url, a['href'])
```

```
        if full_url not in self.pdf_links:
            self.pdf_links.append({'url': full_url, 'text':
a.text.strip()})

    def is_relevant_to_sustainable_finance(self, text):
        keywords = ['sustainable finance', 'esg', 'green bond', 'social impact',
'environmental impact',
                    'climate risk', 'sustainability report', 'corporate
responsibility']
        return any(keyword in text.lower() for keyword in keywords)

    def attempt_crawl(self, url, user_agent):
        for _ in range(self.max_retries):
            try:
                response = self.session.get(url,
headers=self.get_headers(user_agent), timeout=10)
                if response.status_code == 200:
                    print(f"Successfully crawled: {url}")
                    if response.headers.get('Content-Type', '').startswith('text/
html'):
                        self.extract_esg_data(url, response.text)
                    elif response.headers.get('Content-Type',
'').startswith('application/pdf'):
                        self.save_pdf(url, response.content)
                    return True
                else:
                    print(f"Failed to crawl {url}: HTTP {response.status_code}")
            except requests.RequestException as e:
                print(f"Error crawling {url} with {user_agent}: {e}")

                self.delay()
            return False

    def crawl_url(self, url):
        if not self.can_fetch(url, self.desktop_user_agent):
            print(f"Robots.txt disallows desktop user agent: {url}")
            if self.can_fetch(url, self.mobile_user_agent):
                print(f"Attempting with mobile user agent: {url}")
                return self.attempt_crawl(url, self.mobile_user_agent)
            else:
                print(f"Robots.txt disallows both user agents: {url}")
                return False
```

```
        return self.attempt_crawl(url, self.desktop_user_agent)

def crawl(self, max_pages=100):
    self.setup()

    if not self.urls_to_crawl:
        self.urls_to_crawl.append(self.start_url)

    pages_crawled = 0
    while self.urls_to_crawl and pages_crawled < max_pages:
        url = self.urls_to_crawl.popleft()
        if url not in self.crawled_urls:
            if self.crawl_url(url):
                pages_crawled += 1
                self.crawled_urls.add(url)
                self.delay()

    print(f"Crawling complete. Successfully crawled {pages_crawled} pages.")
    self.save_esg_data()
    self.save_news_links()
    self.save_pdf_links()

def save_esg_data(self):
    with open('esg_data.csv', 'w', newline='', encoding='utf-8') as file:
        writer = csv.DictWriter(file, fieldnames=['url', 'environmental',
'social', 'governance'])
        writer.writeheader()
        for data in self.esg_data:
            writer.writerow({
                'url': data['url'],
                'environmental': ', '.join([f"{k}: {v}" for k, v in
data['environmental'].items()]),
                'social': ', '.join([f"{k}: {v}" for k, v in
data['social'].items()]),
                'governance': ', '.join([f"{k}: {v}" for k, v in
data['governance'].items()])
            })
        print("ESG data saved to esg_data.csv")

def save_news_links(self):
    with open('news_links.csv', 'w', newline='', encoding='utf-8') as file:
        writer = csv.DictWriter(file, fieldnames=['url', 'text', 'relevant'])
        writer.writeheader()
```

```
        for news in self.news_links:
            writer.writerow({
                'url': news['url'],
                'text': news['text'],
                'relevant':
self.is_relevant_to_sustainable_finance(news['text'])
            })
        print("News links saved to news_links.csv")

    def save_pdf_links(self):
        # Code for saving PDF in S3 or filesystem

    def save_pdf(self, url, content):
        # Code for saving PDF in S3 or filesystem

# Example usage
if __name__ == "__main__":
    start_url = input("Enter the starting URL to crawl for ESG data and news: ")
    crawler = EnhancedESGCrawler(start_url)
    crawler.crawl(max_pages=50)
```

2. Richten Sie die verschiedenen Attribute ein, darunter Benutzeragenten, leere Sammlungen für URLs und Datenspeicherlisten.
3. Passen Sie die Schlüsselwörter und Relevanzkriterien in der `is_relevant_to_sustainable_finance()` Methode an Ihre spezifischen Bedürfnisse an.
4. Stellen Sie sicher, dass die Datei `robots.txt` das Crawlen der Website zulässt und dass Sie die in der Datei `robots.txt` angegebene Crawling-Verzögerung und den Benutzeragenten verwenden.
5. Erwägen Sie, das bereitgestellte Webcrawler-Skript nach Bedarf für Ihre Organisation wie folgt anzupassen:
  - Implementieren Sie die `fetch_sitemap()` Methode für eine effizientere URL-Erkennung.
  - Verbessern Sie die Fehlerprotokollierung und -behandlung für den Produktionseinsatz.
  - Implementieren Sie eine ausgefeiltere Analyse der Relevanz von Inhalten.
  - Fügen Sie Kontrollen für Tiefe und Breite hinzu, um den Umfang des Crawlings einzuschränken.

# Aufbau der Infrastruktur AWS

Es gibt viele AWS-Services, die Sie zum Aufbau der Web-Crawling-Infrastruktur verwenden können. Der Abschnitt [Architektur](#) dieses Handbuchs enthält einen Lösungsvorschlag. Wir empfehlen Ihnen, beim Aufbau der unterstützenden Infrastruktur für Ihren Webcrawler Folgendes in Betracht AWS-Services zu ziehen:

- [Verwenden Sie Amazon Virtual Private Cloud \(Amazon VPC\), um die VPC und die Subnetze zu erstellen.](#)
- Initiieren Sie den Crawling-Prozess mithilfe von [Amazon EventBridge Scheduler](#).
- [Verwalten Sie die Web-Crawler-Jobs mithilfe von AWS BatchJobs und Job-Warteschlangen.](#)
- Verwenden Sie eine der folgenden Lösungen, um die Webcrawler-Jobs auszuführen:
  - Amazon Elastic Container Service (Amazon ECS) -Container auf [AWS Fargate](#)
  - [Amazon Elastic Compute Cloud \(Amazon EC2\) -Instanzen](#)

## Note

[Wenn Ihre Anwendung Störungen bewältigen kann, sollten Sie die Verwendung von Amazon EC2-Spot-Instances über Spot Fleet in Betracht ziehen.](#) Flotten von Spot-Instances können Ihnen dabei helfen, erheblich an Rechenkosten zu sparen.

- AWS Lambda [Funktionen](#)
- Speichern Sie die abgerufenen Daten und Rohdateien in einem Amazon Simple Storage Service (Amazon S3) [-Bucket](#).

# Bewährte Methoden für ethische Webcrawler

In diesem Abschnitt werden bewährte Verfahren und wichtige ethische Überlegungen zum Aufbau einer Web-Crawling-Anwendung erörtert, die Umwelt-, Sozial- und Unternehmensführungsdaten (ESG) sammelt. Wenn Sie sich an diese bewährten Methoden halten, können Sie Ihr Projekt und Ihre Organisation schützen und zu einem verantwortungsvolleren und nachhaltigeren Web-Ökosystem beitragen. Dieser Ansatz hilft Ihnen, auf wertvolle Daten zuzugreifen und sie für Forschung, Wirtschaft und Innovation auf eine Weise zu nutzen, die alle Beteiligten respektiert.

## Einhaltung von Robots.txt

Die Datei robots.txt wird auf Websites verwendet, um mit Webcrawlern und Bots darüber zu kommunizieren, welche Teile der Website aufgerufen oder gecrawlt werden sollen und welche nicht. Wenn ein Webcrawler auf einer Website auf die Datei robots.txt trifft, analysiert er die Anweisungen und passt sein Crawling-Verhalten entsprechend an. Dadurch wird verhindert, dass der Crawler gegen die Anweisungen des Website-Besitzers verstößt, und es wird eine kooperative Beziehung zwischen der Website und dem Crawler aufrechterhalten. Daher hilft die Datei robots.txt bei der Zugriffskontrolle, beim Schutz sensibler Inhalte, beim Lastmanagement und bei der Einhaltung gesetzlicher Vorschriften.

Wir empfehlen Ihnen, die ACM bewährte Methode für zu befolgen.

- Überprüfen und respektieren Sie immer die Regeln in der Datei robots.txt.
- Bevor Sie eine URL crawlen, überprüfen Sie die Regeln sowohl für Desktop- als auch für mobile Benutzeragenten.
- Wenn die Website nur mobile Benutzeragenten zulässt, verwenden Sie für Ihre Anfrage einen anderen Agent-Header, z. B. einen Mobile-Agent-Header.

Das Fehlen einer Datei robots.txt bedeutet nicht unbedingt, dass Sie eine Website nicht crawlen können oder sollten. Das Crawlen sollte immer verantwortungsbewusst durchgeführt werden, wobei die Ressourcen der Website und die impliziten Rechte des Eigentümers zu respektieren sind. Für den Fall, dass keine robots.txt vorhanden ist, werden die folgenden bewährten Methoden empfohlen:

- Gehen Sie davon aus, dass das Crawlen erlaubt ist, gehen Sie aber vorsichtig vor.
- Wenden Sie höfliche Crawling-Praktiken an.

- Erwägen Sie, den Eigentümer der Website um Erlaubnis zu bitten, wenn Sie vorhaben, ausgiebiges Crawlen durchzuführen.

## Begrenzung der Crawl-Rate

Verwenden Sie eine angemessene Crawling-Rate, um eine Überlastung des Servers zu vermeiden. Implementieren Sie Verzögerungen zwischen Anfragen, entweder wie in der Datei robots.txt angegeben oder indem Sie eine zufällige Verzögerung verwenden. Für kleine oder mittelgroße Websites kann eine Anfrage alle 10—15 Sekunden angemessen sein. Für größere Websites oder Websites mit ausdrücklichen Crawlberechtigungen können 1—2 Anfragen pro Sekunde angemessen sein.

## Transparenz durch Benutzeragenten

Identifizieren Sie Ihren Crawler im User-Agent-Header. Diese HTTP-Header-Informationen dienen dazu, das Gerät zu identifizieren, das den Inhalt anfordert. In der Regel ist das Wort Bot im Namen des Agenten enthalten. Crawler und andere Bots verwenden manchmal ein wichtiges Feld in der Kopfzeile, um Kontaktinformationen hinzuzufügen.

## Effizientes Crawlen

Verwenden Sie die Sitemap, die vom Eigentümer der Website entwickelt wurde, um sich auf wichtige Seiten zu konzentrieren.

## Adaptiver Ansatz

Programmieren Sie den Crawler so, dass er zu einem mobilen Benutzeragenten wechselt, falls die Desktop-Version nicht erfolgreich ist. Dies kann dem Crawler Zugriff gewähren und die Belastung des Servers der Website verringern.

## Fehlerbehandlung

Stellen Sie sicher, dass der Crawler verschiedene HTTP-Statuscodes angemessen behandelt. Der Crawler sollte beispielsweise pausieren, wenn er auf den Statuscode 429 („Zu viele Anfragen“) trifft.

Wenn der Crawler kontinuierlich 403-Statuscodes („Verboten“) empfängt, sollten Sie erwägen, das Crawlen zu beenden.

## Das Crawlen erfolgt stapelweise

Wir empfehlen Folgendes:

- Anstatt alle URLs auf einmal zu crawlen, teilen Sie die Aufgabe in kleinere Batches auf. Dies kann dazu beitragen, die Last zu verteilen und das Risiko von Problemen wie Timeouts oder Ressourcenengpässen zu verringern.
- Wenn davon ausgegangen wird, dass die gesamte Crawling-Aufgabe lange andauern wird, sollten Sie erwägen, sie in mehrere kleinere, leichter zu verwaltende Aufgaben aufzuteilen. Dadurch kann der Prozess skalierbarer und belastbarer werden.
- Wenn die Anzahl der URLs zu durchforstenden Personen relativ gering ist, sollten Sie die Verwendung einer serverlosen Lösung in Betracht ziehen, z. B. AWS Lambda Lambda-Funktionen eignen sich hervorragend für kurzlebige, ereignisgesteuerte Aufgaben, da sie automatisch skalieren und das Ressourcenmanagement übernehmen.

## Sicherheit

Für Web-Crawling-Rechenaufgaben empfehlen wir, die Umgebung so zu konfigurieren, dass nur ausgehender Datenverkehr zugelassen wird. Dies trägt zur Erhöhung der Sicherheit bei, indem die Angriffsfläche minimiert und das Risiko eines unbefugten eingehenden Zugriffs verringert wird. Wenn nur ausgehende Verbindungen zugelassen werden, kann der Crawling-Prozess mit den Ziel-Websites kommunizieren und die erforderlichen Daten abrufen. Außerdem wird jeglicher eingehende Datenverkehr eingeschränkt, der das System potenziell gefährden könnte.

## Weitere Überlegungen

Lesen Sie die folgenden zusätzlichen Überlegungen und bewährten Methoden:

- Suchen Sie in den Nutzungsbedingungen oder Datenschutzbestimmungen der Website nach den Crawling-Richtlinien.
- Suchen Sie im HTML-Code nach meta Tags, die möglicherweise Crawling-Anweisungen enthalten.

- 
- Beachten Sie die gesetzlichen Beschränkungen in Ihrer Gerichtsbarkeit in Bezug auf die Datenerhebung und -nutzung.
  - Seien Sie bereit, das Crawlen zu beenden, wenn Sie vom Eigentümer der Website dazu aufgefordert werden.

## Häufig gestellte Fragen

### Was ist, wenn die Datei robots.txt nicht verfügbar ist?

Das Fehlen einer Datei robots.txt bedeutet nicht unbedingt, dass Sie eine Website nicht crawlen können oder sollten. Das Crawlen sollte immer verantwortungsbewusst durchgeführt werden, wobei die Ressourcen der Website und die impliziten Rechte des Eigentümers der Website zu respektieren sind.

### Was ist, wenn eine Datei sitemaps.xml nicht verfügbar ist?

Je nach Anforderung können Sie einen der folgenden Schritte ausführen:

- Suchen Sie nach HTML-Sitemaps — Suchen Sie nach einer HTML-Sitemap-Seite, auf der die wichtigen Seiten der Website aufgeführt sind. Diese sind oft in der Fußzeile verlinkt.
- Von der Startseite aus crawlen — Beginne mit dem Crawlen von der Startseite aus und folge internen Links, um andere Seiten zu entdecken.
- URL-Muster analysieren — Analysieren Sie die URL-Struktur der Website, um Muster zu identifizieren und programmatisch Potenzial zu generieren. URLs
- Überprüfen Sie die Datei robots.txt — Überprüfen Sie die Datei robots.txt auf unzulässige Seiten oder Verzeichnisse. Diese können Hinweise auf die Seitenstruktur geben.
- API-Endpunkte überprüfen — Einige Websites bieten API-Endpunkte, die zum Abrufen von Inhalts- und Strukturinformationen verwendet werden können.
- Suchmaschinenergebnisse überprüfen — Verwenden Sie Suchmaschinen, um indizierte Seiten der Website mithilfe des [Suchoperators site: zu finden](#), z. B. `site:example.com`
- Backlinks analysieren — Analysieren Sie Backlinks zur Website, um wichtige Seiten zu entdecken, auf die andere Websites verweisen.
- Webarchive überprüfen — Suchen Sie in Internetarchiven wie der [Wayback Machine nach](#) älteren Versionen der Website, die möglicherweise Sitemaps oder andere Strukturen hatten.
- Suchen Sie nach Mustern für das Content Management System (CMS) — Wenn Sie das CMS identifizieren können, verwenden Sie gängige URL-Muster, die mit diesem System verknüpft sind.
- Bestätigen Sie das JavaScript Rendern — Wenn die Website stark davon abhängig ist JavaScript, stellen Sie sicher, dass Ihr Crawler rendern kann JavaScript, um dynamisch geladene Inhalte

zu erkennen. Bei einigen Websites wird die Datei sitemap.xml geladen, nachdem das JavaScript Rendern aktiviert wurde.

## Kann ich anstelle von Amazon EC2 oder Amazon ECS eine serverlose Lösung verwenden?

Ja. [AWS Lambda](#) Funktionen für das Webcrawlen können eine praktikable Option sein, insbesondere für kleinere oder modularere Crawling-Aufgaben. Für groß angelegte, lang andauernde Crawling-Operationen könnte jedoch ein traditionellerer Ansatz besser geeignet sein, der Amazon Elastic Compute Cloud (Amazon EC2) -Instances oder Amazon Elastic Container Service (Amazon ECS) verwendet. Es ist wichtig, dass Sie Ihre spezifischen Anforderungen und Kompromisse sorgfältig abwägen, wenn Sie den richtigen Compute-Service für Ihre Web-Crawling-Anforderungen auswählen.

## Warum erhält der Crawler einen 403-Statuscode?

HTTP 403 ist ein HTTP-Statuscode, der bedeutet, dass der Zugriff auf die angeforderte Ressource verboten ist. Wenn die Anfrage korrekt war, hat der Server die Anfrage verstanden und wird sie nicht erfüllen. Um einen 403-Statuscode zu verhindern, können Sie Folgendes tun:

- Beschränken Sie Ihre Crawling-Rate.
- Prüfen Sie, ob die Sitemap oder die Datei robots.txt dem Crawler den Zugriff auf die URL ermöglicht.
- Versuchen Sie es mit einem mobilen Benutzeragenten anstelle eines Desktop-Benutzeragenten.

Wenn keiner der oben genannten Schritte funktioniert, sollten Sie die Entscheidung der Eigentümer der Website respektieren und die Seite nicht crawlen.

## Nächste Schritte und Ressourcen

Nachdem Sie die Rohdaten zu Umwelt, Soziales und Unternehmensführung (ESG) erfasst haben, können Sie wie folgt vorgehen, um aussagekräftige Informationen aus den Daten zu extrahieren:

1. Daten bereinigen — Die Daten können eine erhebliche Menge irrelevanter Informationen enthalten, die nichts mit ESG-Faktoren und Finanzdaten zu tun haben. Es ist wichtig, diese irrelevanten Daten zu entfernen und nur die Informationen aufzubewahren, die für die Durchführung der erforderlichen Analysen erforderlich sind. Sie können Tools wie [yfinance](#) verwenden, um die Daten zu bereinigen.
2. Extrahieren und transformieren Sie die Daten — Extrahieren Sie die relevanten Merkmale oder Variablen aus den Rohdaten und transformieren Sie sie in ein Format, das für die Analyse geeignet ist. Sie können die Daten zur besseren Lesbarkeit und Übersichtlichkeit in ein tabellarisches Format umwandeln. Sie können eine Bibliothek verwenden, z. B. [pandas](#), um die Daten zu verfeinern. Sie können die Daten auch mithilfe von Feature-Engineering, Datennormalisierung und abgeleiteten Metriken transformieren.
3. Analysen durchführen — Sie können verschiedene analytische Aufgaben ausführen. Dies kann die Generierung deskriptiver Statistiken, die Erstellung von Datenvisualisierungen und die Durchführung explorativer Datenanalysen umfassen, um Einblicke in die ESG-Leistung der Unternehmen zu gewinnen.
4. Wenden Sie maschinelles Lernen an — Sie können die bereinigten und transformierten Daten verwenden, um Modelle für maschinelles Lernen zu trainieren. Mithilfe dieser Modelle können Sie Unternehmen identifizieren, die derzeit finanzielle Nachhaltigkeit nachweisen, und ihre future Nachhaltigkeitsleistung prognostizieren.

Mithilfe des Webcrawlers und dieses Datenauswertungsprozesses können Sie sich effektiv ein umfassendes Bild von den Nachhaltigkeitspraktiken und der finanziellen Leistung der Unternehmen machen, die Sie bewerten. Sie können diese Informationen verwenden, um fundierte Anlageentscheidungen zu treffen, Fortschritte zu verfolgen und nachhaltige Geschäftspraktiken zu unterstützen.

## Ressourcen

- [Was ist ein Webcrawler?](#) (Cloudflare Webseite)
- [Leitfaden für ESG-Investitionen](#) (Investopedia Webseite)

# Tools

- [Beautiful Soup](#) (Beautiful Soup Dokumentation)
- [Umgang mit tabellarischen, relationalen Daten](#) (pandas Webseite)

# Dokumentverlauf

In der folgenden Tabelle werden wichtige Änderungen in diesem Leitfaden beschrieben. Um Benachrichtigungen über zukünftige Aktualisierungen zu erhalten, können Sie einen [RSS-Feed](#) abonnieren.

Änderung	Beschreibung	Datum
<a href="#">Erste Veröffentlichung</a>	—	6. Januar 2025

# AWS Glossar zu präskriptiven Leitlinien

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

## Zahlen

### 7 Rs

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

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

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

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

## A

### ABAC

Siehe [attributbasierte](#) Zugriffskontrolle.

### abstrahierte Dienste

Siehe [Managed Services](#).

### ACID

Siehe [Atomarität, Konsistenz, Isolierung und Haltbarkeit](#).

### Aktiv-Aktiv-Migration

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

### Aktiv-Passiv-Migration

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

### Aggregatfunktion

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

## AI

Siehe [künstliche Intelligenz](#).

## AIOps

Siehe [Operationen im Bereich künstliche Intelligenz](#).

## Anonymisierung

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

## Anti-Muster

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

## Anwendungssteuerung

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

## Anwendungsportfolio

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

## künstliche Intelligenz (KI)

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

## Operationen mit künstlicher Intelligenz (AIOps)

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

## Asymmetrische Verschlüsselung

Ein Verschlüsselungsalgorithmus, der ein Schlüsselpaar, einen öffentlichen Schlüssel für die Verschlüsselung und einen privaten Schlüssel für die Entschlüsselung verwendet. Sie können den öffentlichen Schlüssel teilen, da er nicht für die Entschlüsselung verwendet wird. Der Zugriff auf den privaten Schlüssel sollte jedoch stark eingeschränkt sein.

## Atomizität, Konsistenz, Isolierung, Haltbarkeit (ACID)

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

## Attributbasierte Zugriffskontrolle (ABAC)

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

## autoritative Datenquelle

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

## Availability Zone

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

## AWS Framework für die Einführung der Cloud (AWS CAF)

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

Cloud-Einführung vorzubereiten. Weitere Informationen finden Sie auf der [AWS -CAF-Webseite](#) und dem [AWS -CAF-Whitepaper](#).

### AWS Workload-Qualifizierungsrahmen (AWS WQF)

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

## B

### schlechter Bot

Ein [Bot](#), der Einzelpersonen oder Organisationen stören oder ihnen Schaden zufügen soll.

### BCP

Siehe [Planung der Geschäftskontinuität](#).

### Verhaltensdiagramm

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

### Big-Endian-System

Ein System, welches das höchstwertige Byte zuerst speichert. Siehe auch [Endianness](#).

### Binäre Klassifikation

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

### Bloom-Filter

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

## Blau/Grün-Bereitstellung

Eine Bereitstellungsstrategie, bei der Sie zwei separate, aber identische Umgebungen erstellen. Sie führen die aktuelle Anwendungsversion in einer Umgebung (blau) und die neue Anwendungsversion in der anderen Umgebung (grün) aus. Mit dieser Strategie können Sie schnell und mit minimalen Auswirkungen ein Rollback durchführen.

## Bot

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

## Botnetz

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

## branch

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

## Zugang durch Glasbruch

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

## Brownfield-Strategie

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

## Puffer-Cache

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

## Geschäftsfähigkeit

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

## Planung der Geschäftskontinuität (BCP)

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

# C

## CAF

[Weitere Informationen finden Sie unter Framework AWS für die Cloud-Einführung.](#)

## Bereitstellung auf Kanaren

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

## CCoE

Weitere Informationen finden Sie [im Cloud Center of Excellence](#).

## CDC

Siehe [Erfassung von Änderungsdaten](#).

## Erfassung von Datenänderungen (CDC)

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

## Chaos-Technik

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

## CI/CD

Siehe [Continuous Integration und Continuous Delivery](#).

## Klassifizierung

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

## clientseitige Verschlüsselung

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

## Cloud-Exzellenzzentrum (CCoE)

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

## Cloud Computing

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

## Cloud-Betriebsmodell

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

## Phasen der Einführung der Cloud

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

- Projekt – Durchführung einiger Cloud-bezogener Projekte zu Machbarkeitsnachweisen und zu Lernzwecken
- Fundament — Tätigen Sie grundlegende Investitionen, um Ihre Cloud-Einführung zu skalieren (z. B. Einrichtung einer landing zone, Definition eines CCo E, Einrichtung eines Betriebsmodells)
- Migration – Migrieren einzelner Anwendungen
- Neuentwicklung – Optimierung von Produkten und Services und Innovation in der Cloud

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

## CMDB

Siehe [Datenbank für das Konfigurationsmanagement](#).

## Code-Repository

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

## Kalter Cache

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

## Kalte Daten

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

## Computer Vision (CV)

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

## Drift in der Konfiguration

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

## Verwaltung der Datenbankkonfiguration (CMDB)

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

## Konformitätspaket

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

## Kontinuierliche Bereitstellung und kontinuierliche Integration (CI/CD)

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

## CV

Siehe [Computer Vision](#).

## D

### Daten im Ruhezustand

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

### Datenklassifizierung

Ein Prozess zur Identifizierung und Kategorisierung der Daten in Ihrem Netzwerk auf der Grundlage ihrer Kritikalität und Sensitivität. Sie ist eine wichtige Komponente jeder Strategie für das Management von Cybersecurity-Risiken, da sie Ihnen hilft, die geeigneten Schutz- und Aufbewahrungskontrollen für die Daten zu bestimmen. Die Datenklassifizierung ist ein Bestandteil der Sicherheitssäule im AWS Well-Architected Framework. Weitere Informationen finden Sie unter [Datenklassifizierung](#).

### Datendrift

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

### Daten während der Übertragung

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

### Datennetz

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

### Datenminimierung

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

### Datenperimeter

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

## Vorverarbeitung der Daten

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

## Herkunft der Daten

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

## betreffene Person

Eine Person, deren Daten gesammelt und verarbeitet werden.

## Data Warehouse

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

## Datenbankdefinitionssprache (DDL)

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

## Datenbankmanipulationssprache (DML)

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

## DDL

Siehe [Datenbankdefinitionssprache](#).

## Deep-Ensemble

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

## Deep Learning

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

## defense-in-depth

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

## delegierter Administrator

In AWS Organizations kann ein kompatibler Dienst ein AWS Mitgliedskonto registrieren, um die Konten der Organisation und die Berechtigungen für diesen Dienst zu verwalten. Dieses Konto wird als delegierter Administrator für diesen Service bezeichnet. Weitere Informationen und eine Liste kompatibler Services finden Sie unter [Services, die mit AWS Organizations funktionieren](#) in der AWS Organizations -Dokumentation.

## Einsatz

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

## Entwicklungsumgebung

Siehe [Umgebung](#).

## Detektivische Kontrolle

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

## Abbildung des Wertstroms in der Entwicklung (DVSM)

Ein Prozess zur Identifizierung und Priorisierung von Einschränkungen, die sich negativ auf Geschwindigkeit und Qualität im Lebenszyklus der Softwareentwicklung auswirken. DVSM erweitert den Prozess der Wertstromanalyse, der ursprünglich für Lean-Manufacturing-Praktiken

konzipiert wurde. Es konzentriert sich auf die Schritte und Teams, die erforderlich sind, um durch den Softwareentwicklungsprozess Mehrwert zu schaffen und zu steigern.

## digitaler Zwilling

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

## Maßtabelle

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

## Katastrophe

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

## Notfallwiederherstellung (DR)

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

## DML

Siehe Sprache zur [Datenbankmanipulation](#).

## Domainorientiertes Design

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

## DR

Siehe [Disaster Recovery](#).

### Erkennung von Driften

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

## DVSM

Siehe [Abbildung des Wertstroms in der Entwicklung](#).

## E

### EDA

Siehe [explorative Datenanalyse](#).

### EDI

Siehe [elektronischer Datenaustausch](#).

### Edge-Computing

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

### elektronischer Datenaustausch (EDI)

Der automatisierte Austausch von Geschäftsdokumenten zwischen Organisationen. Weitere Informationen finden Sie unter [Was ist elektronischer Datenaustausch](#).

### Verschlüsselung

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

### Verschlüsselungsschlüssel

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

## Endianismus

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

## Endpunkt

[Siehe](#) Service-Endpunkt.

## Endpunkt-Services

Ein Service, den Sie in einer Virtual Private Cloud (VPC) hosten können, um ihn mit anderen Benutzern zu teilen. Sie können einen Endpunktdienst mit anderen AWS-Konten oder AWS Identity and Access Management (IAM AWS PrivateLink -) Prinzipalen erstellen und diesen Berechtigungen gewähren. Diese Konten oder Prinzipale können sich privat mit Ihrem Endpunktservice verbinden, indem sie Schnittstellen-VPC-Endpunkte erstellen. Weitere Informationen finden Sie unter [Einen Endpunkt-Service erstellen](#) in der Amazon Virtual Private Cloud (Amazon VPC)-Dokumentation.

## Unternehmensressourcenplanung (ERP)

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

## Envelope-Verschlüsselung

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

## Umgebung

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

- **Entwicklungsumgebung** – Eine Instance einer laufenden Anwendung, die nur dem Kernteam zur Verfügung steht, das für die Wartung der Anwendung verantwortlich ist. Entwicklungsumgebungen werden verwendet, um Änderungen zu testen, bevor sie in höhere Umgebungen übertragen werden. Diese Art von Umgebung wird manchmal als Testumgebung bezeichnet.
- **Niedrigere Umgebungen** – Alle Entwicklungsumgebungen für eine Anwendung, z. B. solche, die für erste Builds und Tests verwendet wurden.

- Produktionsumgebung – Eine Instance einer laufenden Anwendung, auf die Endbenutzer zugreifen können. In einer CI/CD Pipeline ist die Produktionsumgebung die letzte Bereitstellungsumgebung.
- Höhere Umgebungen – Alle Umgebungen, auf die auch andere Benutzer als das Kernentwicklungsteam zugreifen können. Dies kann eine Produktionsumgebung, Vorproduktionsumgebungen und Umgebungen für Benutzerakzeptanztests umfassen.

## Epics

In der agilen Methodik sind dies funktionale Kategorien, die Ihnen helfen, Ihre Arbeit zu organisieren und zu priorisieren. Epics bieten eine allgemeine Beschreibung der Anforderungen und Implementierungsaufgaben. Zu den Sicherheitsepen AWS von CAF gehören beispielsweise Identitäts- und Zugriffsmanagement, Detektivkontrollen, Infrastruktursicherheit, Datenschutz und Reaktion auf Vorfälle. Weitere Informationen zu Epics in der AWS -Migrationsstrategie finden Sie im [Leitfaden zur Programm-Implementierung](#).

## ERP

Siehe [Enterprise Resource Planning](#).

## Explorative Datenanalyse (EDA)

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

## F

### Faktentabelle

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

### schnell scheitern

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

## Grenze zur Fehlerisolierung

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

## Feature-Zweig

Siehe [Zweig](#).

## Features

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

## Bedeutung der Feature

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

## Featuretransformation

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

## Eingabeaufforderung mit wenigen Klicks

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

## FGAC

Siehe [detaillierte Zugriffskontrolle](#).

### Feinkörnige Zugriffskontrolle (FGAC)

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

### Flash-Cut-Migration

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

## FM

Siehe [Fundamentmodell](#).

### Fundamentmodell (FM)

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

## G

### Generative KI

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

### Geoblocking

Siehe [geografische Einschränkungen](#).

### Geografische Einschränkungen (Geoblocking)

Bei Amazon eine Option CloudFront, um zu verhindern, dass Benutzer in bestimmten Ländern auf Inhaltsverteilungen zugreifen. Sie können eine Zulassungsliste oder eine Sperrliste verwenden,

um zugelassene und gesperrte Länder anzugeben. Weitere Informationen finden Sie in [der Dokumentation unter Beschränkung der geografischen Verteilung Ihrer Inhalte](#). CloudFront

## Gitflow-Workflow

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

## goldenes Bild

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

## Greenfield-Strategie

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

## Integritätsschutz

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

# H

## HEKTAR

Siehe [Hochverfügbarkeit](#).

## Heterogene Datenbankmigration

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

## hohe Verfügbarkeit (HA)

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

## historische Modernisierung

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

## Daten zurückhalten

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

## Homogene Datenbankmigration

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

## heiße Daten

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

## Hotfix

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

## Hypercare-Phase

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

## I

### IaC

Sehen Sie [Infrastruktur als Code](#).

### Identitätsbasierte Richtlinie

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

### Leerlaufanwendung

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

### IloT

Siehe [Industrielles Internet der Dinge](#).

### unveränderliche Infrastruktur

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

## Eingehende (ingress) VPC

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

## Inkrementelle Migration

Eine Cutover-Strategie, bei der Sie Ihre Anwendung in kleinen Teilen migrieren, anstatt eine einziges vollständiges Cutover durchzuführen. Beispielsweise könnten Sie zunächst nur einige Microservices oder Benutzer auf das neue System umstellen. Nachdem Sie sich vergewissert haben, dass alles ordnungsgemäß funktioniert, können Sie weitere Microservices oder Benutzer schrittweise verschieben, bis Sie Ihr Legacy-System außer Betrieb nehmen können. Diese Strategie reduziert die mit großen Migrationen verbundenen Risiken.

## Industrie 4.0

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

## Infrastruktur

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

## Infrastructure as Code (IaC)

Der Prozess der Bereitstellung und Verwaltung der Infrastruktur einer Anwendung mithilfe einer Reihe von Konfigurationsdateien. IaC soll Ihnen helfen, das Infrastrukturmanagement zu zentralisieren, Ressourcen zu standardisieren und schnell zu skalieren, sodass neue Umgebungen wiederholbar, zuverlässig und konsistent sind.

## industrielles Internet der Dinge (T) Ilo

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

## Inspektions-VPC

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

## Internet of Things (IoT)

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

## Interpretierbarkeit

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

## IoT

Siehe [Internet der Dinge](#).

## IT information library (ITIL, IT-Informationsbibliothek)

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

## T service management (ITSM, IT-Servicemanagement)

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

## BIS

Siehe [IT-Informationsbibliothek](#).

## ITSM

Siehe [IT-Servicemanagement](#).

## L

### Labelbasierte Zugangskontrolle (LBAC)

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

### Landing Zone

Eine landing zone ist eine gut strukturierte AWS Umgebung mit mehreren Konten, die skalierbar und sicher ist. Dies ist ein Ausgangspunkt, von dem aus Ihre Organisationen Workloads und Anwendungen schnell und mit Vertrauen in ihre Sicherheits- und Infrastrukturmgebung starten und bereitstellen können. Weitere Informationen zu Landing Zones finden Sie unter [Einrichtung einer sicheren und skalierbaren AWS -Umgebung mit mehreren Konten..](#)

### großes Sprachmodell (LLM)

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

### Große Migration

Eine Migration von 300 oder mehr Servern.

### SCHWARZ

Siehe [Labelbasierte Zugriffskontrolle](#).

### Geringste Berechtigung

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

### Lift and Shift

Siehe [7 Rs](#).

### Little-Endian-System

Ein System, welches das niedrigwertigste Byte zuerst speichert. Siehe auch [Endianness](#).

## LLM

Siehe [großes Sprachmodell](#).

### Niedrigere Umgebungen

Siehe [Umgebung](#).

## M

### Machine Learning (ML)

Eine Art künstlicher Intelligenz, die Algorithmen und Techniken zur Mustererkennung und zum Lernen verwendet. ML analysiert aufgezeichnete Daten, wie z. B. Daten aus dem Internet der Dinge (IoT), und lernt daraus, um ein statistisches Modell auf der Grundlage von Mustern zu erstellen. Weitere Informationen finden Sie unter [Machine Learning](#).

### Hauptzweig

Siehe [Filiale](#).

### Malware

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

### verwaltete Dienste

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

### Manufacturing Execution System (MES)

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

### MAP

Siehe [Migration Acceleration Program](#).

## Mechanismus

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

## Mitgliedskonto

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

## MES

Siehe [Manufacturing Execution System](#).

## Message Queuing-Telemetrietransport (MQTT)

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

## Microservice

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

## Microservices-Architekturen

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

## Migration Acceleration Program (MAP)

Ein AWS Programm, das Beratung, Unterstützung, Schulungen und Services bietet, um Unternehmen dabei zu unterstützen, eine solide betriebliche Grundlage für die Umstellung auf

die Cloud zu schaffen und die anfänglichen Kosten von Migrationen auszugleichen. MAP umfasst eine Migrationsmethode für die methodische Durchführung von Legacy-Migrationen sowie eine Reihe von Tools zur Automatisierung und Beschleunigung gängiger Migrationsszenarien.

## Migration in großem Maßstab

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

## Migrationsfabrik

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

## Migrationsmetadaten

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

## Migrationsmuster

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

## Migration Portfolio Assessment (MPA)

Ein Online-Tool, das Informationen zur Validierung des Geschäftsszenarios für die Migration auf das bereitstellt. AWS Cloud MPA bietet eine detaillierte Portfoliobewertung (richtige Servergröße, Preisgestaltung, Gesamtbetriebskostenanalyse, Migrationskostenanalyse) sowie Migrationsplanung (Anwendungsdatenanalyse und Datenerfassung, Anwendungsgruppierung,

Migrationspriorisierung und Wellenplanung). Das [MPA-Tool](#) (Anmeldung erforderlich) steht allen AWS Beratern und APN-Partnerberatern kostenlos zur Verfügung.

## Migration Readiness Assessment (MRA)

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

## Migrationsstrategie

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

## ML

[Siehe maschinelles Lernen.](#)

## Modernisierung

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

## Bewertung der Modernisierungsfähigkeit

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

## Monolithische Anwendungen (Monolithen)

Anwendungen, die als ein einziger Service mit eng gekoppelten Prozessen ausgeführt werden. Monolithische Anwendungen haben verschiedene Nachteile. Wenn ein Anwendungs-Feature stark nachgefragt wird, muss die gesamte Architektur skaliert werden. Das Hinzufügen oder

Verbessern der Feature einer monolithischen Anwendung wird ebenfalls komplexer, wenn die Codebasis wächst. Um diese Probleme zu beheben, können Sie eine Microservices-Architektur verwenden. Weitere Informationen finden Sie unter [Zerlegen von Monolithen in Microservices](#).

## MPA

Siehe [Bewertung des Migrationsportfolios](#).

## MQTT

Siehe [Message Queuing-Telemetrietransport](#).

## Mehrklassen-Klassifizierung

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

## veränderbare Infrastruktur

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

## O

### OAC

Siehe [Origin Access Control](#).

### EICHE

Siehe [Zugriffsidentität von Origin](#).

### COM

Siehe [organisatorisches Change-Management](#).

## Offline-Migration

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

## OI

Siehe [Betriebsintegration](#).

## OLA

Siehe Vereinbarung auf [operativer Ebene](#).

## Online-Migration

Eine Migrationsmethode, bei der der Quell-Workload auf das Zielsystem kopiert wird, ohne offline genommen zu werden. Anwendungen, die mit dem Workload verbunden sind, können während der Migration weiterhin funktionieren. Diese Methode beinhaltet keine bis minimale Ausfallzeit und wird in der Regel für kritische Produktionsworkloads verwendet.

## OPC-UA

Siehe [Open Process Communications — Unified Architecture](#).

## Offene Prozesskommunikation — Einheitliche Architektur (OPC-UA)

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

## Vereinbarung auf Betriebsebene (OLA)

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

## Überprüfung der Betriebsbereitschaft (ORR)

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

## Betriebstechnologie (OT)

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

## Betriebsintegration (OI)

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

## Organisationspfad

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

## Organisatorisches Veränderungsmanagement (OCM)

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

## Ursprungszugriffskontrolle (OAC)

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

## Ursprungszugriffsidentität (OAI)

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

## ORR

Weitere Informationen finden Sie unter [Überprüfung der Betriebsbereitschaft](#).

## NICHT

Siehe [Betriebstechnologie](#).

### Ausgehende (egress) VPC

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

## P

### Berechtigungsgrenze

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

### persönlich identifizierbare Informationen (PII)

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

### Personenbezogene Daten

Siehe [persönlich identifizierbare Informationen](#).

### Playbook

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

## PLC

Siehe [programmierbare Logiksteuerung](#).

## PLM

Siehe [Produktlebenszyklusmanagement](#).

## policy

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

## Polyglotte Beharrlichkeit

Unabhängige Auswahl der Datenspeichertechnologie eines Microservices auf der Grundlage von Datenzugriffsmustern und anderen Anforderungen. Wenn Ihre Microservices über dieselbe Datenspeichertechnologie verfügen, kann dies zu Implementierungsproblemen oder zu Leistungseinbußen führen. Microservices lassen sich leichter implementieren und erzielen eine bessere Leistung und Skalierbarkeit, wenn sie den Datenspeicher verwenden, der ihren Anforderungen am besten entspricht.

## Portfoliobewertung

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

## predicate

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

## Prädikat Pushdown

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

## Präventive Kontrolle

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

## Prinzipal

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

## Datenschutz von Natur aus

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

## Privat gehostete Zonen

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

## proaktive Steuerung

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

## Produktlebenszyklusmanagement (PLM)

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

## Produktionsumgebung

Siehe [Umgebung](#).

## Speicherprogrammierbare Steuerung (SPS)

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

## schnelle Verkettung

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

## Pseudonymisierung

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

## publish/subscribe (pub/sub)

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

## Q

### Abfrageplan

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

### Abfrageplanregression

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

# R

## RACI-Matrix

Siehe [verantwortlich, rechenschaftspflichtig, konsultiert, informiert \(RACI\)](#).

## RAG

Siehe Erweiterte [Generierung beim Abrufen](#).

## Ransomware

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

## RASCI-Matrix

Siehe [verantwortlich, rechenschaftspflichtig, konsultiert, informiert \(RACI\)](#).

## RCAC

Siehe [Zugriffskontrolle für Zeilen und Spalten](#).

## Read Replica

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

## neu strukturieren

Siehe [7 Rs](#).

## Recovery Point Objective (RPO)

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

## Wiederherstellungszeitziel (RTO)

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

## Refaktorisierung

Siehe [7 Rs](#).

## Region

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

## Regression

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

## rehosten

Siehe [7 Rs.](#)

## Veröffentlichung

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

## umziehen

Siehe [7 Rs.](#)

## neue Plattform

Siehe [7 Rs.](#)

## Rückkauf

Siehe [7 Rs.](#)

## Ausfallsicherheit

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

## Ressourcenbasierte Richtlinie

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

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

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

## Reaktive Kontrolle

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

## Beibehaltung

Siehe [7 Rs](#).

## zurückziehen

Siehe [7 Rs](#).

## Retrieval Augmented Generation (RAG)

Eine [generative KI-Technologie](#), bei der ein [LLM](#) auf eine maßgebliche Datenquelle verweist, die sich außerhalb seiner Trainingsdatenquellen befindet, bevor eine Antwort generiert wird. Ein RAG-Modell könnte beispielsweise eine semantische Suche in der Wissensdatenbank oder in benutzerdefinierten Daten einer Organisation durchführen. Weitere Informationen finden Sie unter [Was ist RAG](#).

## Drehung

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

## Zugriffskontrolle für Zeilen und Spalten (RCAC)

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

## RPO

Siehe [Recovery Point Objective](#).

## RTO

Siehe [Ziel für die Erholungszeit](#).

## Runbook

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

## S

### SAML 2.0

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

### SCADA

Siehe [Aufsichtskontrolle und Datenerfassung](#).

### SCP

Siehe [Richtlinie zur Dienstkontrolle](#).

### Secret

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

### Sicherheit durch Design

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

## Sicherheitskontrolle

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

## Härtung der Sicherheit

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

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

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

## Automatisierung von Sicherheitsreaktionen

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

## Serverseitige Verschlüsselung

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

## Service-Kontrollrichtlinie (SCP)

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

## Service-Endpoint

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

## Service Level Agreement (SLA)

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

## Service-Level-Indikator (SLI)

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

## Service-Level-Ziel (SLO)

Eine Zielkennzahl, die den Zustand eines Dienstes darstellt, gemessen anhand eines [Service-Level-Indikators](#).

## Modell der geteilten Verantwortung

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

## SIEM

Siehe [Sicherheitsinformations- und Event-Management-System](#).

## Single Point of Failure (SPOF)

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

## SLA

Siehe [Service Level Agreement](#).

## SLI

Siehe [Service-Level-Indikator](#).

## ALSO

Siehe [Service-Level-Ziel](#).

### split-and-seed Modell

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

## SPOTTEN

Siehe [Single Point of Failure](#).

### Sternschema

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

### Strangler-Fig-Muster

Ein Ansatz zur Modernisierung monolithischer Systeme, bei dem die Systemfunktionen schrittweise umgeschrieben und ersetzt werden, bis das Legacy-System außer Betrieb genommen werden kann. Dieses Muster verwendet die Analogie einer Feigenrebe, die zu einem etablierten Baum heranwächst und schließlich ihren Wirt überwindet und ersetzt. Das Muster wurde [eingeführt von Martin Fowler](#) als Möglichkeit, Risiken beim Umschreiben monolithischer Systeme zu managen. Ein Beispiel für die Anwendung dieses Musters finden Sie unter [Schrittweises Modernisieren älterer Microsoft ASP.NET \(ASMX\)-Webservices mithilfe von Containern und Amazon API Gateway](#).

### Subnetz

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

### Aufsichtskontrolle und Datenerfassung (SCADA)

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

## Symmetrische Verschlüsselung

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

## synthetisches Testen

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

## Systemaufforderung

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

## T

### tags

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

### Zielvariable

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

### Aufgabenliste

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

### Testumgebungen

[Siehe Umgebung.](#)

## Training

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

## Transit-Gateway

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

## Stammbasierter Workflow

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

## Vertrauenswürdiger Zugriff

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

## Optimieren

Aspekte Ihres Trainingsprozesses ändern, um die Genauigkeit des ML-Modells zu verbessern. Sie können das ML-Modell z. B. trainieren, indem Sie einen Beschriftungssatz generieren, Beschriftungen hinzufügen und diese Schritte dann mehrmals unter verschiedenen Einstellungen wiederholen, um das Modell zu optimieren.

## Zwei-Pizzen-Team

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

## U

### Unsicherheit

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

### undifferenzierte Aufgaben

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

### höhere Umgebungen

Siehe [Umgebung](#).

## V

### Vacuuming

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

### Versionskontrolle

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

### VPC-Peering

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

## Schwachstelle

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

## W

### Warmer Cache

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

### warme Daten

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

### Fensterfunktion

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

### Workload

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

### Workstream

Funktionsgruppen in einem Migrationsprojekt, die für eine bestimmte Reihe von Aufgaben verantwortlich sind. Jeder Workstream ist unabhängig, unterstützt aber die anderen Workstreams im Projekt. Der Portfolio-Workstream ist beispielsweise für die Priorisierung von Anwendungen, die Wellenplanung und die Erfassung von Migrationsmetadaten verantwortlich. Der Portfolio-Workstream liefert diese Komponenten an den Migrations-Workstream, der dann die Server und Anwendungen migriert.

### WURM

Sehen [Sie einmal schreiben, viele lesen](#).

## WQF

Siehe [AWS Workload-Qualifizierungsrahmen](#).

einmal schreiben, viele lesen (WORM)

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

## Z

### Zero-Day-Exploit

Ein Angriff, in der Regel Malware, der eine [Zero-Day-Sicherheitslücke](#) ausnutzt.

### Zero-Day-Sicherheitslücke

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

### Eingabeaufforderung ohne Zwischenfälle

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

### Zombie-Anwendung

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

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.