



Entwicklerhandbuch

# Amazon Comprehend



# Amazon Comprehend: Entwicklerhandbuch

Copyright © 2024 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 irregeführt 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

Was ist Amazon Comprehend? .....	1
Amazon Comprehend Insights .....	2
Amazon Comprehend Custom .....	2
Flywheels .....	3
Dokumenten-Clustering (Themenmodellierung) .....	3
Beispiele .....	3
Vorteile .....	4
Amazon Comprehend – Preise .....	5
Verwenden Sie Amazon Comprehend zum ersten Mal? .....	5
Funktionsweise .....	7
Insights .....	7
Entitäten .....	8
Ereignisse .....	10
Schlüsselphrasen .....	19
Dominante Sprache .....	20
Stimmung .....	27
Gezielte Stimmung .....	28
Syntaxanalyse .....	45
Amazon Comprehend Custom .....	50
Themenmodellierung .....	50
Modi für die Dokumentenverarbeitung .....	55
Einzeldokumentverarbeitung .....	55
Synchrone Verarbeitung mehrerer Dokumente .....	55
Asynchrone Batchverarbeitung .....	58
Unterstützte Sprachen .....	60
Unterstützte Sprachen .....	60
Sprachen, die von Amazon Comprehend-Funktionen unterstützt werden .....	61
Einrichtung .....	63
Melde dich an für ein AWS-Konto .....	63
Erstellen Sie einen Benutzer mit Administratorzugriff .....	63
Richten Sie das ein AWS CLI .....	65
Erteilen programmgesteuerten Zugriffs .....	65
Erste Schritte .....	68
Verwenden der Konsole .....	70

Analyse in Echtzeit .....	70
Entitäten .....	71
Wichtige Ausdrücke .....	72
Sprache .....	73
Persönlich Identifizierbare Informationen (PII) .....	74
Stimmung .....	76
Gezielte Stimmung .....	77
Syntax .....	79
Analyseaufträge (Konsole) .....	80
Verwenden der API .....	84
Mit SDKs arbeiten AWS .....	84
Echtzeitanalyse (API) .....	85
Erkennen der dominanten Sprache .....	86
Erkennen benannter Entitäten .....	87
Erkennen von Schlüsselphrasen .....	88
Bestimmen der Stimmung .....	89
Echtzeitanalyse für gezielte Stimmung .....	91
Erkennen der Syntax .....	93
Batch-APIs in Echtzeit .....	96
Asynchrone Analyseaufträge (API) .....	101
Amazon Comprehend Insights .....	101
Gezielte Stimmung .....	107
Ereigniserkennung .....	109
Themenmodellierung .....	114
Vertrauen und Sicherheit .....	118
Erkennung der Sichtbarkeit .....	119
Erkennen schädlicher Inhalte mithilfe der API .....	120
Prompt-Sicherheitsklassifizierung .....	123
Prompt-Sicherheitsklassifizierung mithilfe der API .....	123
PII-Erkennung und -Schwärzung .....	125
Persönlich Identifizierbare Informationen (PII) .....	126
Erkennung von PII-Entitäten .....	126
Suchen Sie nach PII-Entitäten .....	127
PII-Entitäten redigieren .....	128
Universelle PII-Entitätstypen .....	128
Länderspezifische PII-Entitätstypen .....	131

Kennzeichnung von PII-Entitäten .....	133
Echtzeitanalyse (Konsole) .....	134
Offsets .....	74
Labels .....	76
Asynchrone Analyseaufträge (Konsole) .....	137
Echtzeitanalyse (API) .....	139
Lokalisieren von PII-Entitäten in Echtzeit (API) .....	140
Kennzeichnung von PII-Echtzeit-Entitäten (API) .....	141
Asynchrone Analyseaufträge (API) .....	142
Auffinden von PII-Entitäten .....	142
PII-Entitäten redigieren .....	147
Dokumentenverarbeitung .....	152
Eingaben für Echtzeitanalysen .....	152
Klartextdokumente .....	153
Semistrukturierte Dokumente .....	153
Bilddateien und gescannte PDF-Dateien .....	153
Amazon-Texttract-Ausgabe .....	153
Maximale Dokumentgrößen für Echtzeitanalysen .....	153
Fehler in halbstrukturierten Dokumenten .....	154
Eingaben für die asynchrone Analyse .....	155
Klartextdokumente .....	156
Semistrukturierte Dokumente .....	156
Bilddateien und gescannte PDF-Dateien .....	157
Amazon-Texttract-Ausgabe-JSON-Dateien .....	157
Festlegen von Optionen für die Textextraktion .....	158
Bewährte Methoden für Images .....	159
Benutzerdefinierte Klassifizierung .....	161
Vorbereitung der Trainingsdaten .....	162
Formate für Trainingsdateien .....	163
Mehrklassenmodus .....	164
Modus mit mehreren Bezeichnungen .....	167
Modelle zur Trainingsklassifizierung .....	170
Trainieren Sie benutzerdefinierte Klassifikatoren (Konsole) .....	172
Trainieren Sie benutzerdefinierte Klassifikatoren (API) .....	177
Testen Sie die Trainingsdaten .....	179
Klassifizieren Sie die Trainingsleistung .....	180

Metriken .....	185
Echtzeitanalyse wird ausgeführt .....	190
Analyse in Echtzeit (Konsole) .....	191
Analyse in Echtzeit (API) .....	193
Ergebnisse für Echtzeitanalysen .....	196
Ausführung von asynchronen Analyseaufträgen .....	198
Eingabedateiformate .....	199
Analyseaufträge (Konsole) .....	200
Analysejobs (API) .....	203
Ausgaben für Analysejobs .....	204
Erkennung benutzerdefinierter Entitäten .....	209
Vorbereitung der Trainingsdaten .....	210
Wann sollten Anmerkungen im Vergleich zu Entitätslisten verwendet werden? .....	211
Entitätslisten .....	212
Anmerkungen .....	215
Trainingserkennungsmodelle .....	229
Trainieren Sie benutzerdefinierte Erkennungsprogramme (Konsole) .....	230
Trainieren von benutzerdefinierten Erkennungen (API) .....	237
Metriken .....	240
Ausführen von Echtzeitanalysen .....	243
Echtzeitanalyse (Konsole) .....	244
Echtzeitanalyse (API) .....	246
Ausgaben für Echtzeitanalysen .....	249
Ausführen asynchroner Analyseaufträge .....	255
Analyseaufträge (Konsole) .....	257
Analyseaufträge (API) .....	259
Ausgaben für Analyseaufträge .....	262
Verwalten von benutzerdefinierten Modellen .....	268
Modell-Versioning mit Amazon Comprehend .....	268
Kopieren von benutzerdefinierten Modellen zwischen AWS-Konten .....	271
Freigeben eines benutzerdefinierten Modells .....	272
Importieren eines benutzerdefinierten Modells .....	282
Flywheels .....	290
Überblick über das Flywheel .....	290
Flywheel-Datensätze .....	291
Flywheel-Erstellung .....	292

Flywheel-Zustände .....	293
Flywheel-Iterationen .....	293
Flywheel-Data-Lakes .....	294
Data-Lake-Ordnerstruktur .....	294
Data-Lake-Management .....	295
IAM-Richtlinien und -Berechtigungen .....	296
Konfigurieren von IAM-Benutzerberechtigungen .....	296
Konfigurieren von Berechtigungen für -AWS KMSSchlüssel .....	297
Erstellen einer Datenzugriffsrolle .....	297
Konfigurieren von Flywheels (Konsole) .....	297
Erstellen eines Flywheel .....	298
Aktualisieren eines Flywheel .....	300
Löschen eines Flywheel .....	301
Konfigurieren von Flywheels (API) .....	301
Erstellen eines Flywheels für ein vorhandenes Modell .....	302
Erstellen eines Flywheels für ein neues Modell .....	302
Beschreiben eines Flywheel .....	303
Aktualisieren eines Flywheel .....	304
Löschen eines Flywheel .....	304
Auflisten der Flywheels .....	305
Konfigurieren von Datensätzen .....	305
Erstellen eines Datensatzes (Konsole) .....	306
Erstellen eines Datensatzes (API) .....	306
Beschreiben eines Datensatzes .....	307
Flywheel-Iterationen .....	308
Iterations-Workflow .....	308
Verwalten von Iterationen (Konsole) .....	309
Verwalten von Iterationen (API) .....	310
Verwenden von Flywheels .....	312
Echtzeitanalyse .....	313
Asynchrone Aufträge .....	313
Verwalten von Endpunkten .....	314
Übersicht über Endpunkte .....	314
Verwenden von Endpunkten .....	315
Überwachen von Endpunkten .....	316
Aktualisieren von Endpunkten .....	319

Verwenden von Trusted Advisor .....	321
Nicht ausgelastete Endpunkte von Amazon Comprehend .....	321
Zugriffsrisiko für Amazon Comprehend-Endpunkte .....	323
Löschen von Endpunkten .....	325
Auto Scaling mit Endpunkten .....	326
Zielverfolgung .....	327
Geplante Skalierung .....	331
Tagging .....	335
Eine neue Ressource taggen .....	336
Anzeigen, Bearbeiten und Löschen von Tags .....	337
Codebeispiele .....	340
Aktionen .....	341
CreateDocumentClassifier .....	342
DeleteDocumentClassifier .....	347
DescribeDocumentClassificationJob .....	349
DescribeDocumentClassifier .....	352
DescribeTopicsDetectionJob .....	355
DetectDominantLanguage .....	357
DetectEntities .....	362
DetectKeyPhrases .....	369
DetectPiiEntities .....	377
DetectSentiment .....	382
DetectSyntax .....	387
ListDocumentClassificationJobs .....	394
ListDocumentClassifiers .....	397
ListTopicsDetectionJobs .....	400
StartDocumentClassificationJob .....	403
StartTopicsDetectionJob .....	407
Szenarien .....	413
Dokumentelemente erkennen .....	413
Führen Sie einen Job zur Themenmodellierung mit Beispieldaten aus .....	419
Trainieren Sie einen benutzerdefinierten Klassifikator und klassifizieren Sie Dokumente .....	424
Serviceübergreifende Beispiele .....	436
Erstellen einer Amazon-Transcribe-Streaming-App .....	437
Einen Amazon Lex Lex-Chatbot erstellen .....	437
Erstellen Sie eine Messaging-Anwendung .....	439



Erstellen einer Anwendung zum Analysieren von Kundenfeedback .....	440
Entitäten in Text erkennen, der aus einem Bild extrahiert wurde .....	446
Sicherheit .....	448
Datenschutz .....	449
KMS-Verschlüsselung in Amazon Comprehend .....	450
Serviceübergreifende Confused-Deputy-Prävention .....	453
Verwenden einer Virtual Private Cloud (VPC) .....	456
VPC-Endpunkte (AWS PrivateLink) .....	463
Identitäts- und Zugriffsverwaltung .....	465
Zielgruppe .....	465
Authentifizierung mit Identitäten .....	466
Verwalten des Zugriffs mit Richtlinien .....	470
So funktioniert Amazon Comprehend mit IAM .....	473
Beispiele für identitätsbasierte Richtlinien .....	481
Von AWS-verwaltete Richtlinien .....	494
Fehlerbehebung .....	498
Protokollieren von Amazon Comprehend-API-Aufrufen mit AWS CloudTrail .....	500
Amazon Comprehend-Informationen in CloudTrail .....	501
Beispiel: Amazon Comprehend-Protokolldateieinträge .....	504
Compliance-Validierung .....	505
Ausfallsicherheit .....	506
Sicherheit der Infrastruktur .....	506
Richtlinien und Kontingente .....	507
Unterstützte Regionen .....	507
Kontingente für integrierte Modelle .....	508
(synchrone) Analyse in Echtzeit .....	508
Asynchrone Analyse .....	510
Kontingente für benutzerdefinierte Modelle .....	513
Allgemeine Kontingente .....	513
Kontingente für Endgeräte .....	514
Klassifizierung von Dokumenten .....	514
Benutzerdefinierte Entitätserkennung .....	519
Kontingente für Schwungräder .....	523
Allgemeine Kontingente für Schwungräder .....	523
Datensatzquoten für benutzerdefinierte Klassifizierungsmodelle .....	524
Datensatzquoten für benutzerdefinierte Modelle zur Erkennung von Entitäten .....	524

---

Tutorials .....	526
Analysieren von Erkenntnissen aus Bewertungen .....	526
Voraussetzungen .....	528
Schritt 1: Dokumente zu Amazon S3 hinzufügen .....	530
Schritt 2: (nur CLI) eine IAM-Rolle erstellen .....	534
Schritt 3: Analyseaufträge ausführen .....	539
Schritt 4: Vorbereiten der Ausgabe .....	543
Schritt 5: Visualisieren der Ausgabe .....	555
Verwenden von S3-Objekt-Lambda-Zugriffspunkten für PII .....	561
Steuern des Zugriffs auf Dokumente mit PII .....	562
Schwärzen von PII aus Dokumenten .....	564
Analysieren von Text mit OpenSearch .....	566
API-Referenz .....	567
Dokumentverlauf .....	568
AWS-Glossar .....	585
.....	dlxxxvi

# Was ist Amazon Comprehend?

Amazon Comprehend verwendet die natürliche Sprachverarbeitung (NLP), um Einblicke in den Inhalt von Dokumenten zu gewinnen. Durch die Erkennung von Einheiten, Schlüsselphrasen, Sprache, Gefühlen und anderen gängigen Elementen eines Dokuments verschafft Amazon Comprehend Einblicke in den Inhalt von Dokumenten. Verwenden Sie Amazon Comprehend, um neue Produkte zu erstellen, die auf dem Verständnis der Struktur von Dokumenten basieren. Mit Amazon Comprehend können Sie beispielsweise Social-Network-Feeds nach Produkterwähnungen durchsuchen oder ein gesamtes Dokument-Repository nach Schlüsselphrasen scannen.

Sie können über die Amazon Comprehend Amazon Comprehend Konsole oder mithilfe der Amazon-Comprehend-APIs auf die Funktionen zur Dokumentenanalyse von Amazon Comprehend zugreifen. Sie können Echtzeitanalysen für kleine Workloads durchführen oder asynchrone Analyseaufträge für große Dokumentsätze starten. Sie können die von Amazon Comprehend bereitgestellten vortrainierten Modelle verwenden oder Ihre eigenen benutzerdefinierten Modelle für Klassifizierung und Entitätserkennung trainieren.

Amazon Comprehend speichert Ihre Inhalte möglicherweise, um die Qualität seiner vortrainierten Modelle kontinuierlich zu verbessern. Weitere Informationen finden Sie unter Häufig gestellte [Fragen zu Amazon Comprehend](#).

Alle Amazon Comprehend-Funktionen akzeptieren UTF-8-Textdokumente als Eingabe. Darüber hinaus akzeptieren die benutzerdefinierte Klassifizierung und die Erkennung benutzerdefinierter Entitäten Bilddateien, PDF-Dateien und Word-Dateien als Eingabe.

Amazon Comprehend kann Dokumente in verschiedenen Sprachen untersuchen und analysieren, abhängig von der spezifischen Funktion. Weitere Informationen finden Sie unter [In Amazon Comprehend unterstützte Sprachen](#). Die [Dominante Sprache](#) Funktion von Amazon Comprehend kann Dokumente untersuchen und die vorherrschende Sprache für eine weitaus breitere Auswahl von Sprachen bestimmen.

## Themen

- [Amazon Comprehend Insights](#)
- [Amazon Comprehend Custom](#)
- [Flywheels](#)
- [Dokumenten-Clustering \(Themenmodellierung\)](#)
- [Beispiele](#)

- [Vorteile](#)
- [Amazon Comprehend – Preise](#)
- [Verwenden Sie Amazon Comprehend zum ersten Mal?](#)

## Amazon Comprehend Insights

Amazon Comprehend verwendet ein vortrainiertes Modell, um ein Dokument oder einen Satz von Dokumenten zu untersuchen und zu analysieren, um Erkenntnisse darüber zu gewinnen. Dieses Modell wird kontinuierlich anhand eines großen Textkörpers trainiert, sodass Sie keine Trainingsdaten bereitstellen müssen.

Amazon Comprehend analysiert die folgenden Arten von Erkenntnissen:

- Entitäten – Verweise auf die Namen von Personen, Orten, Elementen und Speicherorten, die in einem Dokument enthalten sind.
- Schlüsselphrasen – Ausdrücke, die in einem Dokument vorkommen. Beispielsweise könnte ein Dokument über ein Basketball-Spiel die Namen der Teams, den Namen des Veranstaltungsorts und die endgültige Punktzahl zurückgeben.
- Persönlich identifizierbare Informationen (PII) – personenbezogene Daten, die eine Person identifizieren können, z. B. eine Adresse, Kontonummer oder Telefonnummer.
- Sprache – Die vorherrschende Sprache eines Dokuments.
- Stimmung – Die vorherrschende Stimmung eines Dokuments, die positiv, neutral, negativ oder gemischt sein kann.
- Gezielte Stimmung – Die Stimmungen, die bestimmten Entitäten in einem Dokument zugeordnet sind. Die Stimmung für jedes Vorkommen einer Entität kann positiv, negativ, neutral oder gemischt sein.
- Syntax – Die Sprachteile für jedes Wort im Dokument.

Weitere Informationen finden Sie unter [Insights](#).

## Amazon Comprehend Custom

Sie können Amazon Comprehend an Ihre spezifischen Anforderungen anpassen, ohne über die erforderlichen Fähigkeiten verfügen zu müssen, um auf Machine Learning basierende NLP-Lösungen

zu erstellen. Mit automatischem Machine Learning oder AutoML erstellt Amazon Comprehend Custom in Ihrem Namen benutzerdefinierte NLP-Modelle unter Verwendung von Daten, die Sie bereits haben.

**Benutzerdefinierte Klassifizierung** – Erstellen Sie benutzerdefinierte Klassifizierungsmodelle (Klassifikatoren), um Ihre Dokumente in Ihren eigenen Kategorien zu organisieren.

**Benutzerdefinierte Entitätserkennung** – Erstellen Sie benutzerdefinierte Entitätserkennungsmodelle (Erkennungen), die Text nach Ihren spezifischen Begriffen und namensbasierten Ausdrücken analysieren können.

Weitere Informationen finden Sie unter [Amazon Comprehend Custom](#).

## Flywheels

Verwenden Sie Flywheels, um das Training und die Verwaltung benutzerdefinierter Modellversionen im Laufe der Zeit zu vereinfachen. Ein Flywheel hilft bei der Orchestrierung der Aufgaben im Zusammenhang mit dem Training und der Bewertung neuer Versionen eines Modells. Flywheels unterstützen benutzerdefinierte Klartextmodelle für die benutzerdefinierte Klassifizierung und die Erkennung benutzerdefinierter Entitäten. Weitere Informationen finden Sie unter [Flywheels](#).

## Dokumenten-Clustering (Themenmodellierung)

Sie können Amazon Comprehend auch verwenden, um einen Korpus von Dokumenten zu untersuchen und sie basierend auf ähnlichen Schlüsselwörtern darin zu organisieren. Dokumenten-Clustering (Themenmodellierung) ist nützlich, um einen großen Korpus von Dokumenten in Themen oder Clustern zu organisieren, die basierend auf der Wortfrequenz ähnlich sind. Weitere Informationen finden Sie unter [Themenmodellierung](#).

## Beispiele

Die folgenden Beispiele zeigen, wie Sie die Amazon Comprehend-Operationen in Ihren Anwendungen verwenden können.

Example 1: Suchen von Dokumenten zu einem Thema

Suchen Sie die Dokumente zu einem bestimmten Thema mithilfe der Amazon Comprehend-Themenmodellierung. Scannen Sie eine Reihe von Dokumenten, um die besprochenen Themen zu

ermitteln und die Dokumente zu finden, die jedem Thema zugeordnet sind. Sie können die Anzahl der Themen angeben, die Amazon Comprehend aus dem Dokumentsatz zurückgeben soll.

Example 2: Finden Sie heraus, wie Kunden von Ihren Produkten profitieren

Wenn Ihr Unternehmen einen Katalog veröffentlicht, lassen Sie Amazon Comprehend Ihnen mitteilen, was Kunden von Ihren Produkten halten. Senden Sie jeden Kundenkommentar an den DetectSentiment Vorgang und dieser informiert Sie darüber, ob sich Kunden positiv, negativ, neutral oder gemischt zu einem Produkt fühlen.

Example 3: Entdecken Sie, was für Ihre Kunden wichtig ist

Verwenden Sie die Amazon Comprehend-Themenmodellierung, um die Themen zu ermitteln, über die Ihre Kunden in Ihren Foren und Messageboards sprechen, und verwenden Sie dann die Entitätserkennung, um die Personen, Orte und Objekte zu ermitteln, die sie mit dem Thema verknüpfen. Verwenden Sie die Stimmungsanalyse, um festzustellen, wie Ihre Kunden ein Thema bewerten.

## Vorteile

Zu den Vorteilen der Verwendung von Amazon Comprehend gehören:

- Integrieren Sie eine leistungsstarke Verarbeitung natürlicher Sprache in Ihre Apps – Amazon Comprehend eliminiert die Komplexität beim Aufbau von Textanalysefunktionen in Ihre Anwendungen, indem es eine leistungsstarke und genaue Verarbeitung natürlicher Sprache mit einer einfachen API zur Verfügung stellt. Sie benötigen kein Fachwissen in Textanalysen, um die Erkenntnisse zu nutzen, die Amazon Comprehend generiert.
- Deep Learning-basierte natürliche Sprachverarbeitung – Amazon Comprehend verwendet Deep Learning-Technologie, um Text genau zu analysieren. Unsere Modelle werden ständig mit neuen Daten über mehrere Domains hinweg trainiert, um die Genauigkeit zu verbessern.
- Skalierbare Verarbeitung natürlicher Sprache – Mit Amazon Comprehend können Sie Millionen von Dokumenten analysieren, sodass Sie die darin enthaltenen Erkenntnisse entdecken können.
- Integriert mit anderen - AWS Services – Amazon Comprehend ist so konzipiert, dass es nahtlos mit anderen - AWS Services wie Amazon S3 AWS KMS, und funktioniert AWS Lambda. Speichern Sie Ihre Dokumente in Amazon S3 oder analysieren Sie Echtzeitdaten mit Firehose. Die Unterstützung von AWS Identity and Access Management (IAM) erleichtert die sichere Steuerung des Zugriffs

auf Amazon Comprehend-Operationen. Mit IAM können Sie Benutzer und Gruppen erstellen und verwalten, um Ihren Entwicklern und Endbenutzern den entsprechenden Zugriff zu gewähren.

- Verschlüsselung von Ausgabeergebnissen und Volume-Daten – Amazon S3 ermöglicht es Ihnen bereits, Ihre Eingabedokumente zu verschlüsseln, und Amazon Comprehend erweitert dies noch weiter. Mit Ihrem eigenen KMS-Schlüssel können Sie die Ausgabeergebnisse Ihres Auftrags und die Daten auf dem Speichervolume verschlüsseln, das der Rechen-Instance zugeordnet ist, die den Analyseauftrag verarbeitet. Das Ergebnis ist eine deutlich verbesserte Sicherheit.
- Kostengünstig – Bei Amazon Comprehend fallen keine Mindestgebühren oder Vorabverpflichtungen an. Sie zahlen für die Dokumente, die Sie analysieren, und für benutzerdefinierte Modelle, die Sie trainieren.

## Amazon Comprehend – Preise

Mit Amazon Comprehend zahlen Sie nur für die Ressourcen, die Sie tatsächlich nutzen. Wenn Sie ein - AWS Neukunde sind, können Sie kostenlos mit Amazon Comprehend beginnen. Weitere Informationen finden Sie unter [AWS Kostenloses Nutzungskontingent für](#) .

Für die Ausführung von Echtzeit- oder asynchronen Analyseaufträgen fällt eine Nutzungsgebühr an. Sie zahlen für das Training benutzerdefinierter Modelle und für das benutzerdefinierte Modellmanagement. Bei Echtzeitanfragen mit benutzerdefinierten Modellen zahlen Sie für den Endpunkt ab dem Zeitpunkt, an dem Sie Ihren Endpunkt starten, bis zu dem Zeitpunkt, an dem Sie den Endpunkt löschen. Für die Nutzung von Flywheels fallen keine zusätzlichen Gebühren an. Wenn Sie jedoch eine Flywheel-Iteration ausführen, fallen die Standardgebühren für das Training einer neuen Modellversion und das Speichern der Modelldaten an.

Die -Preise und weitere detaillierte Informationen finden Sie unter [Amazon Comprehend – Preise](#).

## Verwenden Sie Amazon Comprehend zum ersten Mal?

Wenn Sie Amazon Comprehend zum ersten Mal verwenden, empfehlen wir Ihnen, der Reihe nach die folgenden Abschnitte zu lesen:

1. [Funktionsweise](#) – In diesem Abschnitt werden Amazon Comprehend-Konzepte vorgestellt.
2. [Einrichtung](#) – In diesem Abschnitt erstellen Sie ein -Konto und richten die ein AWS CLI.
3. [Erste Schritte mit Amazon Comprehend](#) – In diesem Abschnitt führen Sie einen Amazon Comprehend-Analyseauftrag aus.

4. [Tutorial: Analysieren von Erkenntnissen aus Kundenrezensionen mit Amazon Comprehend](#) – In diesem Abschnitt führen Sie Stimmungs- und Entitätsanalysen durch und visualisieren die Ergebnisse.
5. [Amazon Comprehend API Reference](#) – Referenzdokumentation für Amazon Comprehend-Operationen.

AWS stellt die folgenden Ressourcen bereit, um mehr über den Amazon Comprehend-Service zu erfahren:

- Der [AWS Machine Learning Blog](#) enthält nützliche Artikel über Amazon Comprehend .
- [Amazon Comprehend Resources](#) bietet nützliche Videos und Tutorials zu Amazon Comprehend .



# Funktionsweise

Amazon Comprehend verwendet ein vortrainiertes Modell, um Erkenntnisse über ein Dokument oder eine Reihe von Dokumenten zu gewinnen. Dieses Modell wird kontinuierlich anhand eines großen Textkörpers trainiert, sodass Sie keine Trainingsdaten bereitstellen müssen.

Sie können Amazon Comprehend verwenden, um Ihre eigenen benutzerdefinierten Modelle für die benutzerdefinierte Klassifizierung und die Erkennung benutzerdefinierter Entitäten zu erstellen. Sie können verwenden [Flywheels](#), um die benutzerdefinierten Modelle zu verwalten.

Amazon Comprehend bietet Themenmodellierung mit einem integrierten Modell. Die Themenmodellierung untersucht einen Korpus von Dokumenten und organisiert die Dokumente basierend auf ähnlichen Schlüsselwörtern darin.

Amazon Comprehend bietet synchrone und asynchrone Dokumentverarbeitungsmodi. Verwenden Sie den synchronen Modus für die Verarbeitung eines Dokuments oder eines Stapels von bis zu 25 Dokumenten. Verwenden Sie einen asynchronen Auftrag, um eine große Anzahl von Dokumenten zu verarbeiten.

Amazon Comprehend arbeitet mit AWS Key Management Service (AWS KMS), um eine erweiterte Verschlüsselung für Ihre Daten bereitzustellen. Weitere Informationen finden Sie unter [KMS-Verschlüsselung in Amazon Comprehend](#).

Die wichtigsten Konzepte

- [Insights](#)
- [Amazon Comprehend Custom](#)
- [Themenmodellierung](#)
- [Modi für die Dokumentenverarbeitung](#)

## Insights

Amazon Comprehend kann ein Dokument oder einen Satz von Dokumenten analysieren, um Einblicke darüber zu gewinnen. Zu den Erkenntnissen, die Amazon Comprehend über ein Dokument entwickelt, gehören:

- [Entitäten](#) – Amazon Comprehend gibt eine Liste von Entitäten zurück, z. B. Personen, Orte und Standorte, die in einem Dokument identifiziert wurden.

- [Ereignisse](#) – Amazon Comprehend erkennt bestimmte Arten von Ereignissen und zugehörige Details.
- [Schlüsselphrasen](#) – Amazon Comprehend extrahiert Schlüsselphrasen, die in einem Dokument vorkommen. Beispielsweise könnte ein Dokument über ein Basketball-Spiel die Namen der Teams, den Namen des Veranstaltungsorts und die endgültige Punktzahl zurückgeben.
- [Persönlich Identifizierbare Informationen \(PII\)](#) – Amazon Comprehend analysiert Dokumente, um personenbezogene Daten zu erkennen, die eine Person identifizieren, z. B. eine Adresse, Kontonummer oder Telefonnummer.
- [Dominante Sprache](#) – Amazon Comprehend identifiziert die vorherrschende Sprache in einem Dokument. Amazon Comprehend kann 100 Sprachen identifizieren.
- [Stimmung](#) – Amazon Comprehend bestimmt die vorherrschende Stimmung eines Dokuments. Stimmung kann positiv, neutral, negativ oder gemischt sein.
- [Gezieltes Stimmungsbild](#) – Amazon Comprehend bestimmt die Stimmung bestimmter Entitäten, die in einem Dokument erwähnt werden. Die Stimmung jeder Erwähnung kann positiv, neutral, negativ oder gemischt sein.
- [Syntaxanalyse](#) – Amazon Comprehend analysiert jedes Wort in Ihrem Dokument und bestimmt den Sprachteil für das Wort. Beispielsweise wird im Satz „Es ist heute in Puerto verstärkt“, „es“ als Pronom identifiziert, „Verstärken“ wird als Verb und „Glühnis“ als richtiges Substantiv identifiziert.

## Entitäten

Eine Entität ist ein Textverweis auf den eindeutigen Namen eines realen Objekts wie Personen, Orte und kommerzielle Elemente sowie auf präzise Verweise auf Messwerte wie Datum und Mengen.

Im Text „John wurde 2012 auf 1313 Mocking-Leinwand verschoben“ könnte „John“ als erkannt werdenPERSON, „1313 Mocking-Leinwand“ könnte als und LOCATION, „2012“ könnte als erkannt werdenDATE.

Jede Entität hat auch einen Wert, der das Vertrauen angibt, dass Amazon Comprehend den Entitätstyp korrekt erkannt hat. Sie können die Entitäten mit niedrigeren Werten herausfiltern, um das Risiko zu verringern, dass falsche Erkennungen verwendet werden.

In der folgenden Tabelle sind die Entitätstypen aufgeführt.

Typ	Beschreibung
COMMERCIAL_ITEM	Ein Produkten
DATUM	Ein vollständiges Datum (z. B. 11/25/2017), Tag (Dienstag), Monat (Mai) oder Uhrzeit (8:30 Uhr)
EREIGNIS	Ein Ereignis, z. B. ein Telefonie, ein Notebook, eine Wahl usw.
LOCATION	Ein bestimmter Standort, z. B. ein Land, eine Stadt, ein See, ein Gebäude usw.
ORGANIZATION	Große Organisationen, wie eine Regierung, ein Unternehmen, Bol, ein Sportteam usw.
OTHER	Entitäten, die nicht in eine der anderen Entitätskategorien passen
Trichter	Personen, Personengruppen, Spitznamen, fiktive Zeichen
QUANTITY	Ein quantifizierter Betrag, z. B. Währung, Prozentsätze, Zahlen, Bytes usw.
TITLE	Ein offizieller Name, der für jede Erstellung oder Designerarbeit wie Filme, Bücher, Lieder usw. vergeben wird.

Detect-Entitätsoperationen können mit jeder der von Amazon Comprehend unterstützten Primärsprachen ausgeführt werden. Dazu gehört nur die vordefinierte (nicht benutzerdefinierte) Entitätserkennung. Alle Dokumente müssen in derselben Sprache vorliegen.

Sie können eine der folgenden API-Operationen verwenden, um Entitäten in einem Dokument oder einer Reihe von Dokumenten zu erkennen.

- [DetectEntities](#)
- [BatchDetectEntities](#)
- [StartEntitiesDetectionJob](#)

Die Operationen geben eine Liste von [API-Entitätsobjekten](#) zurück, eines für jede Entität im - Dokument. Die BatchDetectEntities Operation gibt eine Liste von Entity Objekten zurück,

eine Liste für jedes Dokument im Batch. Der `StartEntitiesDetectionJob` Vorgang startet einen asynchronen Auftrag, der eine Datei erzeugt, die eine Liste von Entity Objekten für jedes Dokument im Auftrag enthält.

Das folgende Beispiel zeigt die Antwort der `-DetectEntitiesOperation`.

```
{
  "Entities": [
    {
      "Text": "today",
      "Score": 0.97,
      "Type": "DATE",
      "BeginOffset": 14,
      "EndOffset": 19
    },
    {
      "Text": "Seattle",
      "Score": 0.95,
      "Type": "LOCATION",
      "BeginOffset": 23,
      "EndOffset": 30
    }
  ],
  "LanguageCode": "en"
}
```

## Ereignisse

Verwenden Sie die Ereigniserkennung, um Textdokumente auf bestimmte Arten von Ereignissen und ihre zugehörigen Entitäten zu analysieren. Amazon Comprehend unterstützt die Ereigniserkennung für große Sammlungen von Dokumenten mithilfe asynchroner Analyseaufträge. Weitere Informationen zu Ereignissen, einschließlich Beispielaufträgen für die Ereignisanalyse, finden Sie unter [Ankündigung des Starts von Amazon Comprehend Events](#).

### Entitäten

Aus dem Eingabetext extrahiert Amazon Comprehend eine Liste von Entitäten, die sich auf das erkannte Ereignis beziehen. Eine Entität kann ein reales Objekt sein, z. B. eine Person, ein Ort oder ein Standort. Eine Entität kann auch ein Konzept sein, z. B. eine Messung, ein Datum oder eine Menge. Jedes Vorkommen einer Entität wird durch eine Erwähnung identifiziert, bei der es sich um einen Textverweis auf die Entität im Eingabetext handelt. Für jede eindeutige Entität sind

alle Erwähnungen in einer Liste gruppiert. Diese Liste enthält Details zu jedem Speicherort im Eingabetext, an dem die Entität auftritt. Amazon Comprehend erkennt nur die Entitäten, die den unterstützten Ereignistypen zugeordnet sind.

Jede Entität, die einem unterstützten Ereignistyp zugeordnet ist, gibt mit den folgenden zugehörigen Details zurück:

- Hinweise: Details für jedes Vorkommen derselben Entität im Eingabetext.
  - BeginOffset: Ein Zeichenversatz im Eingabetext, der anzeigt, wo die Erwähnung beginnt (das erste Zeichen befindet sich an Position 0).
  - EndOffset: Ein Zeichenversatz im Eingabetext, der anzeigt, wo die Erwähnung endet.
  - Punktzahl : Der Grad des Vertrauens, den Amazon Comprehend in die Genauigkeit des Entitätstyps hat.
  - GroupScore: Der Grad der Zuverlässigkeit von Amazon Comprehend, dass die Erwähnung korrekt gruppiert ist, mit anderen Erwähnungen derselben Entität.
  - Text : Der Text der Entität.
  - Typ : Der Typ der Entität. Informationen zu allen unterstützten Entitätstypen finden Sie unter [Entitätstypen](#).

## Ereignisse

Amazon Comprehend gibt die Liste der Ereignisse (der unterstützten Ereignistypen) zurück, die es im Eingabetext erkennt. Jedes Ereignis gibt mit den folgenden zugehörigen Details zurück:

- Typ : Der Typ des Ereignisses. Informationen zu allen unterstützten Ereignistypen finden Sie unter [Ereignistypen](#).
- Argumente : Eine Liste von Argumenten, die sich auf das erkannte Ereignis beziehen. Ein Argument besteht aus einer Entität, die sich auf das erkannte Ereignis bezieht. Die Rolle des Arguments beschreibt die Beziehung, z. B. wer was getan hat, wo und wann .
  - EntityIndex: Ein Indexwert, der eine Entität aus der Liste der Entitäten identifiziert, die Amazon Comprehend für diese Analyse zurückgegeben hat.
  - Rolle : Der Argumenttyp, der beschreibt, wie die Entität für dieses Argument mit dem Ereignis zusammenhängt. Informationen zu allen unterstützten Argumenttypen finden Sie unter [Argumenttypen](#).
- Punktzahl : Das Maß an Vertrauen, das Amazon Comprehend in die Genauigkeit der Rollenerkennung hat.

- **Auslöser** : Eine Liste von Auslösern für das erkannte Ereignis. Ein Auslöser ist ein einzelnes Wort oder eine einzelne Phrase, die das Auftreten des Ereignisses angibt.
- **BeginOffset**: Ein Zeichenversatz im Eingabetext, der anzeigt, wo der Auslöser beginnt (das erste Zeichen befindet sich an Position 0).
- **EndOffset**: Ein Zeichenversatz im Eingabetext, der anzeigt, wo der Auslöser endet.
- **Punktzahl** : Das Maß an Vertrauen, das Amazon Comprehend in die Genauigkeit der Erkennung hat.
- **Text** : Der Text des Auslösers.
- **GroupScore**: Der Grad der Zuverlässigkeit von Amazon Comprehend, dass der Auslöser korrekt mit anderen Auslösern für dasselbe Ereignis gruppiert ist.
- **Typ** : Der Ereignistyp, den dieser Auslöser angibt.

## Ergebnisformat für Ereignisse erkennen

Wenn Ihr Ereigniserkennungsauftrag abgeschlossen ist, schreibt Amazon Comprehend die Analyseergebnisse in den Amazon S3-Ausgabespeicherort, den Sie beim Starten des Auftrags angegeben haben.

Für jedes erkannte Ereignis enthält die Ausgabe Details im folgenden Format:

```
{
  "Entities": [
    {
      "Mentions": [
        {
          "BeginOffset": number,
          "EndOffset": number,
          "Score": number,
          "GroupScore": number,
          "Text": "string",
          "Type": "string"
        }, ...
      ]
    }, ...
  ],
  "Events": [
    {
      "Type": "string",
      "Arguments": [
```

```

    {
      "EntityIndex": number,
      "Role": "string",
      "Score": number
    }, ...
  ],
  "Triggers": [
    {
      "BeginOffset": number,
      "EndOffset": number,
      "Score": number,
      "Text": "string",
      "GroupScore": number,
      "Type": "string"
    }, ...
  ]
}, ...
]
}

```

## Unterstützte Typen für Entitäten, Ereignisse und Argumente

### Entitätstypen

Typ	Beschreibung
DATUM	Jeder Verweis auf ein Datum oder eine Uhrzeit, unabhängig davon, ob spezifisch oder allgemein.
WALTERUNG	Bauten, Flughafen, Schnellwege, Brücken und andere permanente, von Menschen erstellte Strukturen und echte Bautenverbesserungen.
LOCATION	Physische Standorte wie Straßen, Städte, Bundesstaaten, Länder, Wasserkörper oder geografische Koordinaten.
MONETARY_VALUE	Der Wert von etwas in US- oder einer anderen Währung. Der Wert kann spezifisch oder ungefähr sein.

Typ	Beschreibung
ORGANIZATION	Unternehmen und andere Personengruppen, die durch eine etablierte Organisationsstruktur definiert sind.
Trichter	Die Namen oder Spitznamen von Einzelpersonen oder fiktiven Zeichen.
Bol_TITLE	Jeder Titel, der eine Person beschreibt, bei der es sich in der Regel um eine Kategorie von Arbeitskräften (z. B. CEO) oder um eine ehrwürdige Person (z. B. Bol) handelt.
QUANTITY	Eine Zahl oder ein Wert und die Maßeinheit.
STOCK_CODE	Ein Aktien-Ticker-Symbol, z. B. microSDN, eine International Identification Number (ISIN), ein -Standardverfahren zur Identifizierung von Uniformen (CUSIP) oder eine tägliche offizielle Aktienvergabeliste (SEDOL).

## Ereignistypen

Typ	Beschreibung
BANKRUPTZ	Ein rechtliches Verfahren, bei dem eine Person oder ein Unternehmen ausstehende Kredite nicht zurückzahlen kann.
UMGEBUNG	Tritt auf, wenn ein Mitarbeiter eingestellt, gekündigt oder außer Betrieb genommen wird oder sich anderweitig ändert.
CORPORATE_ACQUISTION	Tritt auf, wenn ein Unternehmen den Besitz der meisten oder aller Anteile oder physische n Komponenten eines anderen Unternehmens



Typ	Beschreibung
	erhält, um die Kontrolle über dieses Unternehmen zu erhalten.
INVESTMENT_GENERAL	Tritt auf, wenn eine Person oder ein Unternehmen eine Komponente mit der Möglichkeit kauft, zukünftiges Einkommen oder Anpreisung zu erzielen.
CORPORATE_MERGER	Tritt auf, wenn zwei oder mehr Unternehmen zusammen eine neue juristische Stelle erstellen .
IPO	Ein anfängliches öffentliches Angebot (IPO) von Anteilen eines privaten Guthabens an die Öffentlichkeit bei einer neuen Aktienemission.
RIGHTS_ISSUE	Eine Gruppe von Rechten, die bestehend en Bewerbern zum Kauf zusätzlicher Aktien, sogenannte Abonnementgarantien, im Verhältnis zu ihren bestehenden Inhabern angeboten werden.
SECONDARY_OFFERING	Ein Angebot an Druckern durch eine Drucker eines Unternehmens.
SHELF_OFFERING	Eine Bereitstellung der microSD and Exchange Commission (SEC), die es einem Aussteller ermöglicht, ein neues Sicherheitsproblem zu registrieren und Teile des Problems über einen bestimmten Zeitraum zu verkaufen, ohne die Sicherheit erneut zu registrieren oder Gebühren zu verursachen. Wird auch als Shelf-Registrierung bezeichnet.
TENDER_OFFERING	Ein Angebot zum Kauf einiger oder aller der Geschäftsanteile eines Unternehmens.

Typ	Beschreibung
STOCK_SPLIT	Tritt auf, wenn der Verwaltungsrät eines Unternehmens die Anzahl der ausstehenden Aktien erhöht, indem mehr Aktien an den aktuellen Arzt ausgegeben werden. Dieses Ereignis gilt auch für umgekehrte Aktienaufteilungen.

## Argumenttypen

### Argumenttypen für BANKRUPTCY

Argumenttyp	Beschreibung
FILER	Die Person oder das Unternehmen, die bzw. das den Ausweis eingereicht hat.
DATUM	Das Datum oder die Uhrzeit der Telefonie.
PLACE	Standort oder Einrichtung, an dem (oder in der Nähe) die Reservierung stattfand.

### Argumenttypen für EMPLOYMENT

Typ	Beschreibung
EMPLOYEE	Die Person, die von einem Unternehmen tätig ist.
EMPLOYEE_TITLE	Der Titel des Mitarbeiters.
EMPLOYER	Die Person oder das Unternehmen, die/das den Mitarbeiter beschäftigt.
START_DATE	Das Startdatum oder die Startzeit der Einrichtung.

Typ	Beschreibung
END_DATE	Das Enddatum oder die Endzeit der Einrichtung.

#### Argumenttypen für CORPORATE\_ACQUISITION, INVESTMENT\_GENERAL

Typ	Beschreibung
Eine Telefonie	Der der Transaktion zugeordnete Geldwert.
INVESTEE	Die Person oder das Unternehmen, die/das mit der Investition verbunden ist.
ANWENDEN	Die Person oder das Unternehmen, die/das in die Komponente investieren.
DATUM	Das Datum oder die Uhrzeit der Übernahme oder Anlage.
PLACE	Ort, an dem (oder in der Nähe) die Übernahme oder Investition stattfand.

#### Argumenttypen für CORPORATE\_MERGER

Typ	Beschreibung
DATUM	Das Datum oder die Uhrzeit der Fusion.
NEU_EIDE	Die neue juristische Stelle, die sich aus der Fusion ergibt.
PARTICIPANT	Das Unternehmen hat an der Fusion beteiligt.

## Argumenttypen für IPO, RIGHTS\_ISSUE, SECONDARY\_OFFERING, SHELF\_OFFERING, TENDER\_OFFERING

Typ	Beschreibung
EXPIRE_DATE	Das Ablaufdatum oder die Ablaufzeit des Angebots.
ANWENDEN	Die Person oder das Unternehmen, die/das in die Komponente investieren.
AUSGEBIETEN	Die Person oder das Unternehmen, die/das das Angebot erhält.
OFFERING_AISSION	Der dem Angebot zugeordnete Geldwert.
AUSBIETEN_DATUM	Das Datum oder die Uhrzeit des Angebots.
-Angebot	Die Person oder das Unternehmen, die/das das Angebot initiiert.
OFFEROR_TOTAL_VALUE	Der gesamte monetäre Wert im Zusammenhang mit dem Angebot.
RECORD_DATE	Das Datum oder die Uhrzeit des Angebots.
SELLING_AGENT	Die Person oder das Unternehmen, die/das den Verkauf des Angebots unterstützt hat.
SHARE_PRICE	Der dem Aktienpreis zugeordnete Geldwert.
SHARE_QUANTITY	Die Anzahl der dem Angebot zugeordneten Freigaben.
UNERWRITER	Das Unternehmen, das mit dem Unterschreiben des Angebots verbunden ist.

## Argumenttypen für STOCK\_SPLIT

Typ	Beschreibung
Ker	Das Unternehmen, das die Aktienaufteilung ausgibt.
DATUM	Das Datum oder die Uhrzeit der Aktienaufteilung.
SPLIT_RATIO	Das Verhältnis der erhöhten neuen Anzahl ausstehender Anteile zur aktuellen Anzahl von Anteilen vor der Aktienaufteilung.

## Schlüsselphrasen

Eine Schlüsselphrase ist eine Zeichenfolge, die eine Substantivphrase enthält, die ein bestimmtes Objekt beschreibt. Es besteht im Allgemeinen aus einem Substantiv und den Modifikatoren, die es unterscheiden. Zum Beispiel ist „Tag“ ein Substantiv; „ein toller Tag“ ist ein Substantivausdruck, der einen Artikel („a“) und ein Adjektiv („beautiful“) enthält. Jede Schlüsselphrase enthält eine Punktzahl, die das Vertrauen angibt, das Amazon Comprehend hat, dass es sich bei der Zeichenfolge um eine Substantivphrase handelt. Sie können den Wert verwenden, um festzustellen, ob die Erkennung eine ausreichend hohe Zuverlässigkeit für Ihre Anwendung aufweist.

Detect Key Phrase-Operationen können mit jeder der von Amazon Comprehend unterstützten Primärsprachen durchgeführt werden. Alle Dokumente müssen in derselben Sprache vorliegen.

Sie können eine der folgenden Operationen verwenden, um Schlüsselphrasen in einem Dokument oder einer Reihe von Dokumenten zu erkennen.

- [DetectKeyPhrases](#)
- [BatchDetectKeyPhrases](#)
- [StartKeyPhrasesDetectionJob](#)

Die Operationen geben eine Liste von [KeyPhrase](#) Objekten zurück, eines für jede Schlüsselphrase im Dokument. Die `BatchDetectKeyPhrases` Operation gibt eine Liste von `KeyPhrase` Objekten zurück, eines für jedes Dokument im Batch. Der `StartKeyPhrasesDetectionJob` Vorgang startet

einen asynchronen Auftrag, der eine Datei erzeugt, die eine Liste von KeyPhrase Objekten für jedes Dokument im Auftrag enthält.

Das folgende Beispiel zeigt die Antwort der `-DetectKeyPhrasesOperation`.

```
{
  "LanguageCode": "en",
  "KeyPhrases": [
    {
      "Text": "today",
      "Score": 0.89,
      "BeginOffset": 14,
      "EndOffset": 19
    },
    {
      "Text": "Seattle",
      "Score": 0.91,
      "BeginOffset": 23,
      "EndOffset": 30
    }
  ]
}
```

## Dominante Sprache

Sie können Amazon Comprehend verwenden, um Text zu untersuchen, um die vorherrschende Sprache zu bestimmen. Amazon Comprehend identifiziert die Sprache mithilfe von Kennungen aus RFC 5646 – wenn es eine zweistellige ISO 639-1-Kennung mit einem regionalen Untertag gibt, wird diese verwendet. Andernfalls wird der Code ISO 639-2 aus drei Buchstaben verwendet.

Weitere Informationen zu RFC 5646 finden Sie unter [Tags zur Identifizierung von Sprachen](#) auf der Website der IETF-Tools.

Die Antwort enthält eine Punktzahl, die das Konfidenzniveau angibt, das Amazon Comprehend hat, dass eine bestimmte Sprache die dominierende Sprache im Dokument ist. Jeder Wert ist unabhängig von den anderen Werten. Der Wert gibt nicht an, dass eine Sprache einen bestimmten Prozentsatz eines Dokuments ausmacht.

Wenn ein langes Dokument (z. B. ein Buch) mehrere Sprachen enthält, können Sie das lange Dokument in kleinere Teile aufteilen und die `DetectDominantLanguage` Operation für die

einzelnen Teile ausführen. Anschließend können Sie die Ergebnisse aggregieren, um den Prozentsatz jeder Sprache im längeren Dokument zu bestimmen.

Die Spracherkennung von Amazon Comprehend hat die folgenden Einschränkungen:

- Es unterstützt keine phonetische Spracherkennung. Beispielsweise erkennt es „Arigino“ nicht als Japanisch oder „Nihao“ als Chinesisch.
- Es kann Schwierigkeiten haben, nahe Sprachpaare wie Indonesisch und Malaiisch oder Bosnisch, Italienisch und Serbisch zu unterscheiden.
- Um optimale Ergebnisse zu erzielen, geben Sie mindestens 20 Zeichen Eingabetext an.

Amazon Comprehend erkennt die folgenden Sprachen.

Code	Sprache
af	Afrikaans
am	Amharisch
ar	Arabisch
as	Assam
az	Aserbaidshanisch
ba	Baschkirisch
be	Belarussisch
bn	Bengalisch
bs	Bosnisch
bg	Bulgarisch
ca	Katalanisch
ceb	Cebuano
cs	Tschechisch

Code	Sprache
cv	Chumbash
cy	Walisisch
da	Dänisch
de	Deutsch
el	Griechisch
en	Englisch
eo	Esperanto
et	Estnisch
eu	Baskisch
fa	Persisch
fi	Finnisch
fr	Französisch
gd	Estlisch-griechisch
ga	Trichter
gl	Galizisch
gu	Gujarati
ht	Haitianisch
he	Hebräisch
ha	Hausa
hi	Hindi



Code	Sprache
hr	Kroatisch
hu	Ungarisch
hy	Armenisch
ilo	Iloko
id	Indonesisch
is	Isländisch
it	Italienisch
jv	Javanesisch
ja	Japanisch
kn	Kannada
ka	Georgisch
kk	Kasachisch
km	Zentraler Khmer
ky	Kirghiz
ko	Koreanisch
ku	Poldisch
lo	Lao
la	Lateinisch
lv	Lettisch
lt	Litauisch

Code	Sprache
lb	Luxemburgisch
ml	Malayalam
mt	Maltesisch
mr	Marathi
mk	Mazedonisch
mg	Malagasie
mn	Mongolisch
ms	Malaiisch
my	Burmesisch
ne	Nepali
new	Newari
nl	Niederländisch
no	Norwegisch
or	Orija
om	Oromo
pa	Pandschabi
pl	Polnisch
pt	Portugiesisch
ps	Pushto
qu	Quechua

Code	Sprache
ro	Rumänisch
ru	Russisch
sa	Sanskrit
si	Singhalesisch
sk	Slowakisch
sl	Slowenisch
sd	Sindhi
so	Somali
es	Spanisch
sq	Albanisch
sr	Serbisch
su	Sundanesisch
sw	Swahili
sv	Schwedisch
ta	Tamil
tt	Tatarisch
te	Telugu
tg	Tajik
tl	Tagalog
th	Thailändisch

Code	Sprache
tk	Turkmen
tr	Türkisch
ug	Uighur
uk	Ukrainisch
ur	Urdu
uz	Usbekisch
vi	Vietnamesisch
yi	Yiddisch
yo	Yorgien
zh	Chinesisch (vereinfacht)
zh-TW	Chinesisch (traditionell)

Sie können eine der folgenden Operationen verwenden, um die vorherrschende Sprache in einem Dokument oder einer Reihe von Dokumenten zu erkennen.

- [DetectDominantLanguage](#)
- [BatchDetectDominantLanguage](#)
- [StartDominantLanguageDetectionJob](#)

Die `-DetectDominantLanguageOperation` gibt ein `-DominantLanguage` Objekt zurück. Die `BatchDetectDominantLanguage` Operation gibt eine Liste von `DominantLanguage` Objekten zurück, eines für jedes Dokument im Batch. Der `StartDominantLanguageDetectionJob` Vorgang startet einen asynchronen Auftrag, der eine Datei mit einer Liste von `DominantLanguage` Objekten erzeugt, eines für jedes Dokument im Auftrag.

Das folgende Beispiel zeigt die Antwort der `-DetectDominantLanguageOperation`.

```
{
  "Languages": [
    {
      "LanguageCode": "en",
      "Score": 0.9793661236763
    }
  ]
}
```

## Stimmung

Verwenden Sie Amazon Comprehend, um die Stimmung von Inhalten in UTF-8-kodierten Textdokumenten zu ermitteln. Sie können beispielsweise die Stimmungsanalyse verwenden, um die Stimmungen von Kommentaren in einem Blogbeitrag zu ermitteln, um festzustellen, ob Ihren Lesern der Beitrag gefallen hat.

Sie können die Stimmung für Dokumente in einer der von Amazon Comprehend unterstützten Primärsprachen bestimmen. Alle Dokumente in einem Auftrag müssen sich in derselben Sprache befinden.

Die Stimmungsermittlung gibt die folgenden Werte zurück:

- positiv – Der Text drückt eine insgesamt positive Stimmung aus.
- negativ – Der Text drückt eine insgesamt negative Stimmung aus.
- Gemischt – Der Text drückt sowohl positive als auch negative Stimmungen aus.
- Leerlauf – Der Text drückt weder positive noch negative Stimmungen aus.

Sie können eine der folgenden API-Operationen verwenden, um die Stimmung eines Dokuments oder einer Reihe von Dokumenten zu erkennen.

- [DetectSentiment](#)
- [BatchDetectSentiment](#)
- [StartSentimentDetectionJob](#)

Die Operationen geben die wahrscheinlichste Stimmung für den Text und die Ergebnisse für jede der Stimmungen zurück. Die Punktzahl stellt die Wahrscheinlichkeit dar, dass die Stimmung korrekt erkannt wurde. Im folgenden Beispiel ist es beispielsweise zu 95 Prozent wahrscheinlich, dass der

Text eine Positive Stimmung hat. Es besteht eine Wahrscheinlichkeit von weniger als 1 Prozent, dass der Text eine Negative Stimmung hat. Sie können die verwenden `SentimentScore`, um festzustellen, ob die Genauigkeit der Erkennung den Anforderungen Ihrer Anwendung entspricht.

Die `DetectSentiment` Operation gibt ein Objekt zurück, das die erkannte Stimmung und ein [SentimentScore](#) Objekt enthält. Die `BatchDetectSentiment` Operation gibt eine Liste von Stimmungen und `SentimentScore` Objekten zurück, eines für jedes Dokument im Batch. Der `StartSentimentDetectionJob` Vorgang startet einen asynchronen Auftrag, der eine Datei mit einer Liste von Stimmungen und `SentimentScore` Objekten erzeugt, eines für jedes Dokument im Auftrag.

Das folgende Beispiel zeigt die Antwort der `-DetectSentimentOperation`.

```
{
  "SentimentScore": {
    "Mixed": 0.030585512690246105,
    "Positive": 0.94992071056365967,
    "Neutral": 0.0141543131828308,
    "Negative": 0.00893945890665054
  },
  "Sentiment": "POSITIVE",
  "LanguageCode": "en"
}
```

## Gezielte Stimmung

Gezielte Stimmung bietet ein detailliertes Verständnis der Stimmungen, die bestimmten Entitäten (wie Marken oder Produkten) in Ihren Eingabedokumenten zugeordnet sind.

Der Unterschied zwischen gezielter Stimmung und [Stimmung](#) ist der Grad der Granularität in den Ausgabedaten. Die Stimmungsanalyse bestimmt die vorherrschende Stimmung für jedes Eingabedokument, stellt jedoch keine Daten für die weitere Analyse bereit. Die gezielte Stimmungsanalyse bestimmt die Stimmung auf Entitätsebene für bestimmte Entitäten in jedem Eingabedokument. Sie können die Ausgabedaten analysieren, um die spezifischen Produkte und Services zu ermitteln, die positives oder negatives Feedback erhalten.

Beispielsweise bietet ein Kunde in einer Reihe von Unterhaltungen die folgende Bewertung: „Die Takos waren ansteigend und die Mitarbeiter waren freundlich“. Die Analyse dieser Überprüfung führt zu den folgenden Ergebnissen:

- Die Stimmungsanalyse bestimmt, ob die Gesamtstimmung der einzelnen Lebensmittel positiv, negativ, neutral oder gemischt ist. In diesem Beispiel ist die Gesamtstimmung positiv.
- Die gezielte Stimmungsanalyse bestimmt die Stimmung für Entitäten und Attribute des Studios, die Kunden in den Bewertungen erwähnt haben. In diesem Beispiel machte der Kunde positive Kommentare zu „Tacos“ und „Mitarbeiter“.

Die gezielte Stimmung liefert die folgenden Ausgaben für jeden Analyseauftrag:

- Identität der in den Dokumenten genannten Entitäten.
- Klassifizierung des Entitätstyps für jede Entitätserwähnung.
- Die Stimmung und ein Stimmungswert für jede Entitätserwähnung.
- Gruppen von Erwähnungen (Co-Referenzgruppen), die einer einzelnen Entität entsprechen.

Sie können die [???Konsole](#) oder die [API](#) verwenden, um eine gezielte Stimmungsanalyse durchzuführen. Die Konsole und die API unterstützen Echtzeitanalysen und asynchrone Analysen für gezielte Stimmungen.

Amazon Comprehend unterstützt gezielte Stimmungen für Dokumente in englischer Sprache.

Weitere Informationen zu gezielten Stimmungen, einschließlich eines Tutorials, finden Sie unter [Extrahieren von granularen Stimmungen im Text mit Amazon Comprehend Targeted Sentiment](#) im - Blog für AWS Machine Learning.

## Themen

- [Entitätstypen](#)
- [Co-Referenzgruppe](#)
- [Organisation der Ausgabedatei](#)
- [Echtzeitanalyse mit der Konsole](#)
- [Beispiel für eine gezielte Stimmungsausgabe](#)

## Entitätstypen

Die gezielte Stimmung identifiziert die folgenden Entitätstypen. Es weist den Entitätstyp Telefonie zu, wenn die Entität nicht zu einer anderen Kategorie gehört. Jede Entität, die in der Ausgabedatei erwähnt wird, enthält den Entitätstyp, z. B. "Type": "PERSON".

## Definitionen des Entitätstyps

Entitätstyp	Definition
Trichter	Beispiele hierfür sind Einzelpersonen, Personengruppen, Spitznamen, fiktive Zeichen und Arztnamen.
LOCATION	Geografische Standorte wie Länder, Städte, Bundesstaaten, Adressen, geologische Formationen, Gewässer, natürliche Statuen und astronomische Standorte.
ORGANIZATION	Beispiele hierfür sind Regierungen, Unternehmen, Sportteams und Jungfern.
BOILILITY	Bauten, Flughafen, Hochwege, Brücken und andere permanente, von Menschen erstellte Strukturen und echte Bautenverbesserungen.
Telefonie	Organisation, Gruppe oder Produzent eines bestimmten kommerziellen Artikels oder Produktbereichs.
COMMERCIAL_ITEM	Alle nicht generischen Artikel, einschließlich Fahrzeuge, und große Produkte, für die nur ein Artikel produziert wurde.
MOVIE	Eine Film- oder Telefoniesendung. Entität könnte der vollständige Name, ein Spitzname oder ein Untertitel sein.
MUSIK	Ein Song, vollständig oder teilweise. Außerdem Sammlungen einzelner Musikerstellungen, z. B. ein Album oder eine Anthologie.
BOOK	Ein Buch, das fachlich oder selbstveröffentlicht veröffentlicht wird.
Trichter	Ein offiziell veröffentlichtes Softwareprodukt.
GAME	Ein Spiel, wie z. B. Video-, Bord-, gängige Spiele oder Sport.
PERSONAL_TASTE	Offizielle Titel und Anerkennungen wie Bol, PhD oder Dr.
EREIGNIS	Beispiele hierfür sind , Telefonie, Wahlen, Konflikte, Konferenzen und Werbeveranstaltungen.



Entitätstyp	Definition
DATUM	Jeder Verweis auf ein Datum oder eine Uhrzeit, unabhängig davon, ob spezifisch oder allgemein, ob absolute oder relative.
QUANTITY	Alle Messungen zusammen mit ihren Einheiten (Währung, Prozentsatz, Zahl, Bytes usw.).
ATTRIBUTE	Ein Attribut, Merkmal oder Merkmal einer Entität, z. B. die „Qualität“ eines Produkts, der „Preis“ eines Telefons oder die „Geschwindigkeit“ einer CPU.
OTHER	Entitäten, die keiner der anderen Kategorien angehören.

## Co-Referenzgruppe

Die gezielte Stimmung identifiziert Co-Referenzgruppen in jedem Eingabedokument. Eine Co-Referenzgruppe ist eine Gruppe von Erwähnungen in einem Dokument, die einer realen Entität entsprechen.

### Example


Im folgenden Beispiel einer Kundenbewertung ist „spa“ die Entität mit dem Entitätstyp FACILITY. Die Entität hat zwei zusätzliche Erwähnungen als Pronom („sie“).



## Organisation der Ausgabedatei

Der gezielte Stimmungsanalyseauftrag erstellt eine JSON-Textausgabedatei. Die Datei enthält ein JSON-Objekt für jedes der Eingabedokumente. Jedes JSON-Objekt enthält die folgenden Felder:

- Entitäten – Ein Array von Entitäten, die im Dokument gefunden wurden.
- Datei – Der Dateiname des Eingabedokuments.
- Zeile – Wenn die Eingabedatei ein Dokument pro Zeile ist, enthält Entitäten die Zeilennummer des Dokuments in der Datei.

 Note

Wenn die gezielte Stimmung keine Entitäten im Eingabetext identifiziert, gibt sie ein leeres Array als Ergebnis der Entitäten zurück.

Das folgende Beispiel zeigt Entitäten für eine Eingabedatei mit drei Eingabezeilen. Das Eingabeformat ist ONE\_DOC\_PER\_LINE, daher ist jede Eingabezeile ein Dokument.

```
{ "Entities":[
  {entityA},
  {entityB},
  {entityC}
],
"File": "TargetSentimentInputDocs.txt",
"Line": 0
}
{ "Entities": [
  {entityD},
  {entityE}
],
"File": "TargetSentimentInputDocs.txt",
"Line": 1
}
{ "Entities": [
  {entityF},
  {entityG}
],
"File": "TargetSentimentInputDocs.txt",
"Line": 2
}
```

Eine Entität im Entitäts-Array enthält eine logische Gruppierung (als Co-Referenzgruppe bezeichnet) der Entitätserwähnungen, die im Dokument erkannt wurden. Jede Entität hat die folgende Gesamtstruktur:

```
{
  "DescriptiveMentionIndex": [0],
  "Mentions": [
    {mentionD},
    {mentionE}
  ]
}
```

Eine Entität enthält die folgenden Felder:

- Hinweise – Eine Reihe von Erwähnungen der Entität im Dokument. Das Array stellt eine Co-Referenzgruppe dar. Ein Beispiel finden Sie unter [the section called “Co-Referenzgruppe”](#). Die Reihenfolge der Erwähnungen im Array Belieben ist die Reihenfolge ihres Standorts (Offset) im Dokument. Jede Erwähnung umfasst den Stimmungswert und den Gruppenwert für diese Erwähnung. Der Gruppenwert gibt das Konfidenzniveau an, dass diese Erwähnungen derselben Entität angehören.
- DescriptiveMentionIndex – Ein oder mehrere Indizes in das Array Beliebungen, das den besten Namen für die Entitätsgruppe bereitstellt. Beispielsweise könnte eine Entität drei Erwähnungen mit den Text werten „ABC Ker“, „ABC Ker“ und „it“ haben. Der beste Name ist „ABC Ker“, das den DescriptiveMentionIndex Wert [0,1] hat.

Jede Erwähnung enthält die folgenden Felder

- BeginOffset – Der Offset in den Dokumenttext, in dem die Erwähnung beginnt.
- EndOffset – Der Offset in den Dokumenttext, in dem die Erwähnung endet.
- GroupScore – Das Vertrauen, dass sich alle in der Gruppe genannten Entitäten auf dieselbe Entität beziehen.
- Text – Der Text in dem Dokument, das die Entität identifiziert.
- Typ – Der Typ der Entität. Amazon Comprehend unterstützt eine Vielzahl von [Entitätstypen](#).
- Ergebnis – Modellzuverlässigkeit, dass die Entität relevant ist. Der Wertebereich ist Null bis Eins, wobei einer der sichersten ist.
- MentionSentiment – Enthält die Stimmung und den Stimmungswert für die Erwähnung.

- **Stimmung** – Die Stimmung der Erwähnung. Zu den Werten gehören: POSITIV, NEUTRAL, NEGATIV und GEMISCHT.
- **SentimentScore** – Bietet Modellzuverlässigkeit für jede der möglichen Stimmungen. Der Wertebereich ist Null bis Eins, wobei einer der sichersten ist.

Die Stimmungswerte haben die folgende Bedeutung:

- **positiv** – Die Entitätserwähnung drückt eine positive Stimmung aus.
- **negativ** – Die Entitätserwähnung drückt eine negative Stimmung aus.
- **Gemischt** – Die Entitätserwähnung drückt sowohl positive als auch negative Stimmungen aus.
- **Leerlauf** – Die Entitätserwähnung drückt weder positive noch negative Stimmungen aus.

Im folgenden Beispiel hat eine Entität nur eine Erwähnung im Eingabedokument, daher **DescriptiveMentionIndex** ist Null (die erste Erwähnung im Array „Beliebenschäften“). Die identifizierte Entität ist ein Bol mit dem Namen „I“. Der Stimmungswert ist neutral.

```
{
  "Entities": [
    {
      "DescriptiveMentionIndex": [0],
      "Mentions": [
        {
          "BeginOffset": 0,
          "EndOffset": 1,
          "Score": 0.999997,
          "GroupScore": 1,
          "Text": "I",
          "Type": "PERSON",
          "MentionSentiment": {
            "Sentiment": "NEUTRAL",
            "SentimentScore": {
              "Mixed": 0,
              "Negative": 0,
              "Neutral": 1,
              "Positive": 0
            }
          }
        }
      ]
    }
  ]
}
```

```
],  
  "File": "Input.txt",  
  "Line": 0  
}
```

## Echtzeitanalyse mit der Konsole

Sie können die Amazon Comprehend-Konsole verwenden, um [the section called “Gezielte Stimmung”](#) in Echtzeit auszuführen. Verwenden Sie den Beispieltext oder fügen Sie Ihren eigenen Text in das Eingabetextfeld ein und wählen Sie dann Analysieren aus.

Im Bereich Insights zeigt die Konsole drei Ansichten der gezielten Stimmungsanalyse an:

- **Analysierter Text** – Zeigt den analysierten Text an und unterstrich jede Entität. Die Farbe des Unterstrichs gibt den Stimmungswert (positiv, neutral, negativ oder gemischt) an, den die Analyse der Entität zugewiesen hat. Die Konsole zeigt die Farbzusammenordnungen in der oberen rechten Ecke des analysierten Textfeldes an. Wenn Sie den Mauszeiger über eine Entität bewegen, zeigt die Konsole ein Popup-Bedienfeld mit Analysewerten (Entitätstyp, Stimmungswert) für die Entität an.
- **Ergebnisse** – Zeigt eine Tabelle mit einer Zeile für jede Entitätserwähnung an, die im Text identifiziert wurde. Für jede Entität zeigt die Tabelle die [Entität](#) und den Entitätswert an. Die Zeile enthält auch die primäre Stimmung und den Wert für jeden Stimmungswert. Wenn es mehrere Erwähnungen derselben Entität gibt, die als bezeichnet werden [the section called “Co-Referenzgruppe”](#), zeigt die Tabelle diese Erwähnungen als zusammenklappbaren Satz von Zeilen an, die der Hauptentität zugeordnet sind.

Wenn Sie den Mauszeiger über eine Entitätszeile in der Tabelle Ergebnisse bewegen, hebt die Konsole die Entitätserwähnung im Textfeld Analysiert hervor.

- **Anwendungsintegration** – Zeigt die Parameterwerte der API-Anforderung und die Struktur des JSON-Objekts an, das in der API-Antwort zurückgegeben wird. Eine Beschreibung der Felder im JSON-Objekt finden Sie unter [the section called “Organisation der Ausgabedatei”](#).

## Beispiel für eine Echtzeitanalyse der Konsole

In diesem Beispiel wird der folgende Text als Eingabe verwendet. Dabei handelt es sich um den Standardeingabetext, den die Konsole bereitstellt.

```
Hello Zhang Wei, I am John. Your AnyCompany Financial Services, LLC credit card account  
1111-0000-1111-0008 has a minimum payment
```

of \$24.53 that is due by July 31st. Based on your autopay settings, we will withdraw your payment on the due date from your bank account number XXXXXX1111 with the routing number XXXXX0000.

Customer feedback for Sunshine Spa, 123 Main St, Anywhere. Send comments to Alice at sunspa@mail.com.

I enjoyed visiting the spa. It was very comfortable but it was also very expensive. The amenities were ok but the service made the spa a great experience.

Das Textfeld Analysiert zeigt die folgende Ausgabe für dieses Beispiel. Bewegen Sie die Maus über den Text Zhang Wei, um das Popup-Bedienfeld für diese Entität anzuzeigen.

**Insights Info**

Entities | Key phrases | Language | PII | Sentiment | **Targeted sentiment** | Syntax

**Analyzed text** 
■ Positive 
 ■ Neutral 
 ■ Negative 
 ■ Mixed

Hello Zhang Wei, I am John. Your AnyCompany Financial Services, LLC credit card account 1111-0000-1111-0008 has a minimum payment of \$24.53 that is due by July 31st. Based on your autopay settings, we will withdraw your payment on the due date from your bank account number XXXXXX1111 with the routing number XXXXX0000.

Customer feedback for Sunshine Spa, 123 Main St, Anywhere. Send comments to Alice at sunspa@mail.com.

I enjoyed visiting the spa. It was very comfortable but it was also very expensive. The amenities were ok but the service made the spa a great experience.

Entity type: PERSON ×  
 Entity confidence: 0.99+  
 Sentiment: NEUTRAL  
 Sentiment confidence: 0.99+  
 Total related entities: 5

Die Tabelle Ergebnisse enthält zusätzliche Details zu jeder Entität, einschließlich der Entitätsbewertung, der primären Stimmung und der Punktzahl für jede Stimmung.

## ▼ Results

Entity	Entity type	Entity score	Primary sentiment	Positive score	Ne
⊕ Zhang Wei (5)	PERSON	-	— NEUTRAL	-	-
⊕ John (3)	PERSON	-	— NEUTRAL	-	-
⊕ AnyCompany Financial Services, LLC (2)	ORGANIZATION	-	— NEUTRAL	-	-
credit card account	OTHER	0.99+	— NEUTRAL	0.00	0.0
\$24.53	QUANTITY	0.99+	— NEUTRAL	0.00	0.0
⊕ by July 31st (3)	DATE	-	— NEUTRAL	-	-
bank account	OTHER	0.99+	— NEUTRAL	0.00	0.0
XXXXXX1111	OTHER	0.51	— NEUTRAL	0.00	0.0
Customer	PERSON	0.98	— NEUTRAL	0	0
⊕ Sunshine Spa (5)	FACILITY	-	— MIXED	-	-

In unserem Beispiel erkennt die gezielte Stimmungsanalyse, dass jede Erwähnung Ihres im Eingabetext ein Verweis auf die Personseinheit Zhang Wei ist. Die Konsole zeigt diese Erwähnungen als Satz zusammenklappbarer Zeilen an, die der Hauptentität zugeordnet sind.

## ▼ Results

Entity	Entity type	Entity score	Primary sentiment	Positive score	Ne
⊖ Zhang Wei (5)	PERSON	-	— NEUTRAL	-	-
your	PERSON	0.99+	— NEUTRAL	0	0
your	PERSON	0.67	— NEUTRAL	0	0
Your	ORGANIZATION	0.94	— NEUTRAL	0	0
your	PERSON	0.99+	— NEUTRAL	0	0
Zhang Wei	PERSON	0.99+	— NEUTRAL	0.00	0

Im Bereich Anwendungsintegration wird das JSON-Objekt angezeigt, das die DetectTargetedSentiment API generiert. Ein vollständiges Beispiel finden Sie im folgenden Abschnitt.

## Beispiel für eine gezielte Stimmungsausgabe

Das folgende Beispiel zeigt die Ausgabedatei aus einem gezielten Stimmungsanalyseauftrag. Die Eingabedatei besteht aus drei einfachen Dokumenten:

```
The burger was very flavorful and the burger bun was excellent. However, customer
service was slow.
My burger was good, and it was warm. The burger had plenty of toppings.
The burger was cooked perfectly but it was cold. The service was OK.
```

Die gezielte Stimmungsanalyse dieser Eingabedatei erzeugt die folgende Ausgabe.

```
{"Entities":[
  {
    "DescriptiveMentionIndex": [
      0
    ],
    "Mentions": [
      {
        "BeginOffset": 4,
        "EndOffset": 10,
        "Score": 0.999991,
        "GroupScore": 1,
        "Text": "burger",
        "Type": "OTHER",
        "MentionSentiment": {
          "Sentiment": "POSITIVE",
          "SentimentScore": {
            "Mixed": 0,
            "Negative": 0,
            "Neutral": 0,
            "Positive": 1
          }
        }
      }
    ]
  },
  {
    "DescriptiveMentionIndex": [
      0
    ],
    "Mentions": [
      {
```



```
    "BeginOffset": 38,
    "EndOffset": 44,
    "Score": 1,
    "GroupScore": 1,
    "Text": "burger",
    "Type": "OTHER",
    "MentionSentiment": {
      "Sentiment": "NEUTRAL",
      "SentimentScore": {
        "Mixed": 0.000005,
        "Negative": 0.000005,
        "Neutral": 0.999591,
        "Positive": 0.000398
      }
    }
  }
]
},
{
  "DescriptiveMentionIndex": [
    0
  ],
  "Mentions": [
    {
      "BeginOffset": 45,
      "EndOffset": 48,
      "Score": 0.961575,
      "GroupScore": 1,
      "Text": "bun",
      "Type": "OTHER",
      "MentionSentiment": {
        "Sentiment": "POSITIVE",
        "SentimentScore": {
          "Mixed": 0.000327,
          "Negative": 0.000286,
          "Neutral": 0.050269,
          "Positive": 0.949118
        }
      }
    }
  ]
}
],
{
  "DescriptiveMentionIndex": [
```

```
    0
  ],
  "Mentions": [
    {
      "BeginOffset": 73,
      "EndOffset": 89,
      "Score": 0.999988,
      "GroupScore": 1,
      "Text": "customer service",
      "Type": "ATTRIBUTE",
      "MentionSentiment": {
        "Sentiment": "NEGATIVE",
        "SentimentScore": {
          "Mixed": 0.000001,
          "Negative": 0.999976,
          "Neutral": 0.000017,
          "Positive": 0.000006
        }
      }
    }
  ]
}
],
"File": "TargetSentimentInputDocs.txt",
"Line": 0
}
{
  "Entities": [
    {
      "DescriptiveMentionIndex": [
        0
      ],
      "Mentions": [
        {
          "BeginOffset": 0,
          "EndOffset": 2,
          "Score": 0.99995,
          "GroupScore": 1,
          "Text": "My",
          "Type": "PERSON",
          "MentionSentiment": {
            "Sentiment": "NEUTRAL",
            "SentimentScore": {
              "Mixed": 0,
```

```
        "Negative": 0,
        "Neutral": 1,
        "Positive": 0
    }
}
]
},
{
  "DescriptiveMentionIndex": [
    0,
    2
  ],
  "Mentions": [
    {
      "BeginOffset": 3,
      "EndOffset": 9,
      "Score": 0.999999,
      "GroupScore": 1,
      "Text": "burger",
      "Type": "OTHER",
      "MentionSentiment": {
        "Sentiment": "POSITIVE",
        "SentimentScore": {
          "Mixed": 0.000002,
          "Negative": 0.000001,
          "Neutral": 0.000003,
          "Positive": 0.999994
        }
      }
    },
    {
      "BeginOffset": 24,
      "EndOffset": 26,
      "Score": 0.999756,
      "GroupScore": 0.999314,
      "Text": "it",
      "Type": "OTHER",
      "MentionSentiment": {
        "Sentiment": "POSITIVE",
        "SentimentScore": {
          "Mixed": 0,
          "Negative": 0.000003,
          "Neutral": 0.000006,
```

```
        "Positive": 0.999991
      }
    }
  },
  {
    "BeginOffset": 41,
    "EndOffset": 47,
    "Score": 1,
    "GroupScore": 0.531342,
    "Text": "burger",
    "Type": "OTHER",
    "MentionSentiment": {
      "Sentiment": "POSITIVE",
      "SentimentScore": {
        "Mixed": 0.000215,
        "Negative": 0.000094,
        "Neutral": 0.00008,
        "Positive": 0.999611
      }
    }
  }
]
},
{
  "DescriptiveMentionIndex": [
    0
  ],
  "Mentions": [
    {
      "BeginOffset": 52,
      "EndOffset": 58,
      "Score": 0.965462,
      "GroupScore": 1,
      "Text": "plenty",
      "Type": "QUANTITY",
      "MentionSentiment": {
        "Sentiment": "NEUTRAL",
        "SentimentScore": {
          "Mixed": 0,
          "Negative": 0,
          "Neutral": 1,
          "Positive": 0
        }
      }
    }
  ]
}
```

```
    }
  ]
},
{
  "DescriptiveMentionIndex": [
    0
  ],
  "Mentions": [
    {
      "BeginOffset": 62,
      "EndOffset": 70,
      "Score": 0.998353,
      "GroupScore": 1,
      "Text": "toppings",
      "Type": "OTHER",
      "MentionSentiment": {
        "Sentiment": "NEUTRAL",
        "SentimentScore": {
          "Mixed": 0,
          "Negative": 0,
          "Neutral": 0.999964,
          "Positive": 0.000036
        }
      }
    }
  ]
}
],
"File": "TargetSentimentInputDocs.txt",
"Line": 1
}
{
  "Entities": [
    {
      "DescriptiveMentionIndex": [
        0
      ],
      "Mentions": [
        {
          "BeginOffset": 4,
          "EndOffset": 10,
          "Score": 1,
          "GroupScore": 1,
          "Text": "burger",
```

```
    "Type": "OTHER",
    "MentionSentiment": {
      "Sentiment": "POSITIVE",
      "SentimentScore": {
        "Mixed": 0.001515,
        "Negative": 0.000822,
        "Neutral": 0.000243,
        "Positive": 0.99742
      }
    }
  },
  {
    "BeginOffset": 36,
    "EndOffset": 38,
    "Score": 0.999843,
    "GroupScore": 0.999661,
    "Text": "it",
    "Type": "OTHER",
    "MentionSentiment": {
      "Sentiment": "NEGATIVE",
      "SentimentScore": {
        "Mixed": 0,
        "Negative": 0.999996,
        "Neutral": 0.000004,
        "Positive": 0
      }
    }
  }
]
},
{
  "DescriptiveMentionIndex": [
    0
  ],
  "Mentions": [
    {
      "BeginOffset": 53,
      "EndOffset": 60,
      "Score": 1,
      "GroupScore": 1,
      "Text": "service",
      "Type": "ATTRIBUTE",
      "MentionSentiment": {
        "Sentiment": "NEUTRAL",
```

```
        "SentimentScore": {
          "Mixed": 0.000033,
          "Negative": 0.000089,
          "Neutral": 0.993325,
          "Positive": 0.006553
        }
      }
    ]
  }
],
"File": "TargetSentimentInputDocs.txt",
"Line": 2
}
}
```

## Syntaxanalyse

Verwenden Sie die Syntaxanalyse, um die Wörter aus dem Dokument zu analysieren und den Sprachteil oder die syntaktische Funktion für jedes Wort im Dokument zurückzugeben. Sie können die Substantive, Verben, Adjektive usw. in Ihrem Dokument identifizieren. Verwenden Sie diese Informationen, um den Inhalt Ihrer Dokumente besser zu verstehen und die Beziehung der Wörter im Dokument zu verstehen.

Sie können beispielsweise in einem Dokument nach den Substantiven und dann nach den Verben suchen, die sich auf diese Substantive beziehen. In einem Satz wie „Meine Großmutter hat ihre Couch verschoben“ sehen Sie die Substantive, „Urgroßmutter“ und „Couch“ und das Verb „Verschiebt“. Sie können diese Informationen verwenden, um Anwendungen für die Textanalyse für Wortkombinationen zu erstellen, die Sie interessieren.

Um die Analyse zu starten, analysiert Amazon Comprehend den Quelltext, um die einzelnen Wörter im Text zu finden. Nachdem der Text analysiert wurde, wird jedem Wort der Sprachteil zugewiesen, den es im Quelltext aufnimmt.

Amazon Comprehend kann die folgenden Sprachteile identifizieren.

Token	Sprachteil
ADJ	Adjektiv

Token	Sprachteil Wörter, die normalerweise Substantive ändern.
ADP	Adposition Der Kopf einer präpositionalen oder postpositionalen Phrase.
Adv	Adverb Wörter, die Verben normalerweise ändern. Sie können auch Adjective und andere Adverbs ändern.
EinUX	Hilfsprogramm Funktionswörter, die das Verb einer Verb-Phrase unterstützen.
KCONJ	Koordinierungsverbindung Eine koordinierende Verbindung verbindet Wörter, Wortgruppen oder Klauseln in einem Satz, ohne eine untereinander zu ordnen.
CONJ	Konjunktion Eine Verbindung verbindet Wörter, Wortgruppen oder Klauseln in einem Satz.



Token	Sprachteil
DET	<p>-Ermittler</p> <p>Artikel und andere Wörter, die eine bestimmte Wortgruppe angeben.</p>
INTJ	<p>Interjizieren</p> <p>Wörter, die als Ausrufezeichen oder Teil eines Ausrufezeichens verwendet werden.</p>
NUN	<p>Substantiv</p> <p>Wörter, die eine Person, einen Ort, ein Objekt, einen Arzt oder eine Idee angeben.</p>
NUM	<p>Numerisch</p> <p>Wörter, in der Regel Bestimmte, Adjektive oder Pronomen, die eine Zahl ausdrücken.</p>
O	<p>Sonstige</p> <p>Wörter, denen kein Teil der Sprachkategorie zugewiesen werden kann.</p>
PART	<p>Partikel</p> <p>Funktionswörter, die mit einem anderen Wort oder einer anderen Phrase verknüpft sind, um die Bedeutung zu unterteilen.</p>

Token	Sprachteil
PRON	<p>Pronom</p> <p>Wörter, die Substantive oder Substantive ersetzen.</p>
PROPN	<p>Ordnungsgemäßes Substantiv</p> <p>Ein Substantiv, das der Name einer bestimmten Person, eines bestimmten Orts oder eines bestimmten Objekts ist.</p>
PUNCT	<p>Interpunktion</p> <p>Nicht alphabetische Zeichen, die Text trennen.</p>
SCONJ	<p>Subordinierungsverbindung</p> <p>Eine Verbindung, die eine abhängige Klausel mit einem Satz verbindet. Ein Beispiel für eine untergeordnete Verbindung ist „verursachen“.</p>
SYM	<p>Symbol</p> <p>Wörterähnliche Entitäten wie das Dollarzeichen (\$) oder mathematische Symbole.</p>
VERB	<p>Verb</p> <p>Wörter, die Ereignisse und Aktionen signalisieren.</p>

Weitere Informationen zu den Sprachteilen finden Sie unter [Universal POS Tags](#) auf der Website Universal Dependencies.

Die Operationen geben Token zurück, die das Wort und den Sprachteil identifizieren, den das Wort im Text darstellt. Jedes Token steht für ein Wort im Quelltext. Es gibt die Position des Wortes in der Quelle, den Sprachteil, den das Wort im Text aufnimmt, das Vertrauen, dass Amazon Comprehend hat, dass der Sprachteil korrekt identifiziert wurde, und das Wort, das aus dem Quelltext analysiert wurde.

Im Folgenden finden Sie die Struktur der Liste der Syntax-Token. Für jedes Wort im Dokument wird ein Syntax-Token generiert.

```
{
  "SyntaxTokens": [
    {
      "BeginOffset": number,
      "EndOffset": number,
      "PartOfSpeech": {
        "Score": number,
        "Tag": "string"
      },
      "Text": "string",
      "TokenId": number
    }
  ]
}
```

Jedes Token enthält die folgenden Informationen:

- **BeginOffset** und **EndOffset**– Gibt die Position des Wortes im Eingabetext an.
- **PartOfSpeech**– Stellt zwei Informationen bereit, die den Sprachteil identifiziert **Tag**, und die **Score**, die das Vertrauen darstellt, dass die Amazon Comprehend Syntax den Sprachteil korrekt identifiziert hat.
- **Text**– Gibt das identifizierte Wort an.
- **TokenId**– Stellt eine Kennung für das Token bereit. Die Kennung ist die Position des Tokens in der Liste der Token.

# Amazon Comprehend Custom

Sie können Amazon Comprehend an Ihre spezifischen Anforderungen anpassen, ohne über die erforderlichen Fähigkeiten verfügen zu müssen, um auf Machine Learning basierende NLP-Lösungen zu erstellen. Mithilfe von automatischem Machine Learning oder AutoML erstellt Comprehend Custom in Ihrem Namen benutzerdefinierte NLP-Modelle unter Verwendung der von Ihnen bereitgestellten Trainingsdaten.

Verarbeitung von Eingabedokumenten – Amazon Comprehend unterstützt die einstufige Dokumentenverarbeitung für benutzerdefinierte Klassifizierung und Erkennung benutzerdefinierter Entitäten. Sie können beispielsweise eine Mischung aus Klartextdokumenten und halbstrukturierten Dokumenten (wie PDF-Dokumenten, Microsoft Word-Dokumenten und Bildern) in einen benutzerdefinierten Analyseauftrag eingeben. Weitere Informationen finden Sie unter [Dokumentenverarbeitung](#).

Benutzerdefinierte Klassifizierung – Erstellen Sie benutzerdefinierte Klassifizierungsmodelle (Klassifikatoren), um Ihre Dokumente in Ihren eigenen Kategorien zu organisieren. Geben Sie für jede Klassifizierungsbezeichnung eine Reihe von Dokumenten an, die diese Bezeichnung am besten darstellen, und trainieren Sie Ihren Classifier darauf. Nach dem Training kann ein Classifier für eine beliebige Anzahl unbeschrifteter Dokumentsätze verwendet werden. Sie können die -Konsole für ein codeloses Erlebnis verwenden oder das neueste AWS SDK installieren. Weitere Informationen finden Sie unter [Benutzerdefinierte Klassifizierung](#).

Benutzerdefinierte Entitätserkennung – Erstellen Sie benutzerdefinierte Entitätserkennungsmodelle (Erkennungen), die Text nach Ihren spezifischen Begriffen und namensbasierten Ausdrücken analysieren können. Sie können Erkenner darin trainieren, Begriffe wie Richtliniennummern oder Ausdrücke zu extrahieren, die eine Kundeneskalation implizieren. Um das Modell zu trainieren, stellen Sie eine Liste der Entitäten und eine Reihe von Dokumenten bereit, die sie enthalten. Sobald das Modell trainiert wurde, können Sie Analyseaufträge für es einreichen, um seine benutzerdefinierten Entitäten zu extrahieren. Weitere Informationen finden Sie unter [Erkennung benutzerdefinierter Entitäten](#).

## Themenmodellierung

Sie können Amazon Comprehend verwenden, um den Inhalt einer Sammlung von Dokumenten zu untersuchen und gemeinsame Themen zu ermitteln. Sie können Amazon Comprehend beispielsweise eine Sammlung von Nachrichtenartikeln geben, und es bestimmt die Themen wie

Sport, Richtlinie oder Sport. Der Text in den Dokumenten muss nicht mit Anmerkungen versehen werden.

Amazon Comprehend verwendet ein [auf der Latent-Dirichlet-Zuweisung](#) basierendes Lernmodell, um die Themen in einer Reihe von Dokumenten zu bestimmen. Es untersucht jedes Dokument, um den Kontext und die Bedeutung eines Wortes zu bestimmen. Der Satz von Wörtern, die häufig zum gleichen Kontext über den gesamten Dokumentsatz gehören, bildet ein Thema.

Ein Wort ist einem Thema in einem Dokument zugeordnet, basierend darauf, wie weit dieses Thema in einem Dokument vorherrschend ist und wie viel Affinität das Thema zum Wort hat. Dasselbe Wort kann verschiedenen Themen in verschiedenen Dokumenten zugeordnet werden, basierend auf der Themenverteilung in einem bestimmten Dokument.

Beispielsweise kann das Wort „Gluke“ in einem Artikel, der hauptsächlich über Sport spricht, dem Thema „sports“ zugewiesen werden, während dasselbe Wort in einem Artikel über „Medicine“ dem Thema „Medicine“ zugewiesen wird.

Jedes einem Thema zugeordnete Wort erhält eine Gewichtung, die angibt, wie stark das Wort bei der Definition des Themas hilft. Die Gewichtung gibt an, wie oft das Wort im Thema im Vergleich zu anderen Wörtern im Thema über den gesamten Dokumentsatz hinweg vorkommt.

Für die genauesten Ergebnisse sollten Sie Amazon Comprehend den größtmöglichen Korpus zur Verfügung stellen, mit dem Sie arbeiten können. Um die besten Ergebnisse zu erzielen:

- Sie sollten mindestens 1 000 Dokumente in jedem Themenmodellierungsauftrag verwenden.
- Jedes Dokument sollte mindestens 3 Sätze lang sein.
- Wenn ein Dokument hauptsächlich aus numerischen Daten besteht, sollten Sie es aus dem Korpus entfernen.

Die Themenmodellierung ist ein asynchroner Prozess. Sie senden Ihre Liste der Dokumente mithilfe der [-StartTopicsDetectionJob](#) Operation aus einem Amazon S3-Bucket an Amazon Comprehend. Die Antwort wird an einen Amazon S3-Bucket gesendet. Sie können sowohl den Eingabe- als auch den Ausgabe-Bucket konfigurieren. Rufen Sie eine Liste der Themenmodellierungsaufträge ab, die Sie mit der [-ListTopicsDetectionJobs](#) Operation übermittelt haben, und zeigen Sie Informationen zu einem Auftrag mit der [-DescribeTopicsDetectionJob](#) Operation an. Inhalte, die an Amazon-S3-Buckets geliefert werden, können Kundeninhalte enthalten. Weitere Informationen zum Entfernen sensibler Daten finden Sie unter [Wie entleere ich einen S3 Bucket?](#) oder [Wie lösche ich einen S3 Bucket?](#).

Dokumente müssen in Textdateien im UTF-8-Format vorliegen. Sie können Ihre Dokumente auf zwei Arten einreichen. Die folgende Tabelle zeigt die Optionen.

Format	Beschreibung
Ein Dokument pro Datei	Jede Datei enthält ein Eingabedokument. Dies eignet sich am besten für Sammlungen großer Dokumente.
Ein Dokument pro Zeile	<p>Die Eingabe ist eine einzelne Datei. Jede Zeile in der Datei wird als Dokument betrachtet. Dies eignet sich am besten für kurze Dokumente, z. B. Social-Media-Beiträge.</p> <p>Jede Zeile muss mit einem Zeilenvorschub (LF, \n), einem Zeilenumlauf (CR, \r) oder beidem (CRLF, \r\n) enden. Das Unicode-Zeilentrennzeichen (u+2028) kann nicht verwendet werden, um eine Zeile zu beenden.</p>

Weitere Informationen finden Sie unter [-InputDataConfig](#)Datentyp.

Nachdem Amazon Comprehend Ihre Dokumentensammlung verarbeitet hat, gibt es ein komprimiertes Archiv mit zwei Dateien zurück, `topic-terms.csv` und `doc-topics.csv`. Weitere Informationen zur Ausgabedatei finden Sie unter [OutputDataConfig](#).

Die erste Ausgabedatei, `topic-terms.csv`, ist eine Liste von Themen in der Sammlung. Für jedes Thema enthält die Liste standardmäßig die wichtigsten Begriffe nach Thema entsprechend ihrer Gewichtung. Wenn Sie Amazon Comprehend beispielsweise eine Sammlung von Trichterartikeln geben, könnte es Folgendes zurückgeben, um die ersten beiden Themen in der Sammlung zu beschreiben:

Thema	Begriff	Gewicht
000	Team	0,118533
000	game	0,106072

Thema	Begriff	Gewicht
000	player	0,031625
000	Saison	0.023633
000	spielen	0,02118
000	Fuß	0,024454
000	Kabel	0,016012
000	Spiele	0.016191
000	Fußball	0,015049
000	Quartalsrückstand	0,014239
001	Kabel	0,205236
001	Lebensmittel	0,040686
001	Minuten	0,036062
001	hinzufügen	0,029697
001	Kabel	0,028789
001	Kabel	0,021254
001	Kabel	0,022205
001	Kabel	0,020040
001	Lebensmittel	0,016588
001	Glühbir	0,015101

Die Gewichtungen stellen eine Wahrscheinlichkeitsverteilung über die Wörter in einem bestimmten Thema dar. Da Amazon Comprehend nur die 10 wichtigsten Wörter für jedes Thema zurückgibt,

werden die Gewichtungen nicht auf 1,0 summiert. In den seltenen Fällen, in denen weniger als 10 Wörter in einem Thema enthalten sind, werden die Gewichtungen auf 1,0 summiert.

Die Wörter werden nach ihrer diskriminativen Potenz sortiert, indem sie sich ihr Vorkommen in allen Themen ansehen. In der Regel entspricht dies ihrer Gewichtung, aber in einigen Fällen, z. B. den Wörtern „Play“ und „Höfe“ in der Tabelle, führt dies zu einer Reihenfolge, die nicht mit der Gewichtung identisch ist.

Sie können die Anzahl der zurückzugebenden Themen angeben. Wenn Sie beispielsweise Amazon Comprehend bitten, 25 Themen zurückzugeben, werden die 25 wichtigsten Themen in der Sammlung zurückgegeben. Amazon Comprehend kann bis zu 100 Themen in einer Sammlung erkennen. Wählen Sie die Anzahl der Themen basierend auf Ihrem Wissen über die Domain aus. Es kann einige Experimente dauern, bis die richtige Zahl erreicht ist.

Die zweite Datei, `doc-topics.csv`, listet die einem Thema zugeordneten Dokumente und den Anteil des Dokuments auf, das sich mit dem Thema befasst. Wenn Sie angegeben haben, wird `ONE_DOC_PER_FILE` das Dokument durch den Dateinamen identifiziert. Wenn Sie angegeben haben, wird `ONE_DOC_PER_LINE` das Dokument durch den Dateinamen und die mit 0 indizierte Zeilennummer innerhalb der Datei identifiziert. Amazon Comprehend könnte beispielsweise Folgendes für eine Sammlung von Dokumenten zurückgeben, die mit einem Dokument pro Datei übermittelt werden:

Dokument	Thema	Anteil
sample-doc1	000	0,999330137
sample-doc2	000	0,998532187
sample-doc3	000	0,998384574
...		
sample-docN	000	3.57E-04

Amazon Comprehend verwendet Informationen aus dem Datensatz Lemmatization Lists von MBM , der [hier](#) unter der [Open Database License \(ODbL\) v1.0](#) verfügbar gemacht wird.



# Modi für die Dokumentenverarbeitung

Amazon Comprehend unterstützt drei Dokumentverarbeitungsmodi. Die Wahl des Modus hängt von der Anzahl der Dokumente ab, die Sie verarbeiten müssen, und davon, wie sofort Sie die Ergebnisse anzeigen müssen:

- **Einzeldokument synchron** – Sie rufen Amazon Comprehend mit einem einzigen Dokument auf und erhalten sofort eine synchrone Antwort, die an Ihre Anwendung (oder die Konsole) übermittelt wird.
- **Multi-Document synchronous** – Sie rufen die Amazon Comprehend API mit einer Sammlung von bis zu 25 Dokumenten auf und erhalten eine synchrone Antwort.
- **Asynchroner Batch** – Legen Sie die Dokumente für eine große Sammlung von Dokumenten in einen Amazon S3-Bucket und starten Sie einen asynchronen Auftrag (mit Konsolen- oder API-Operationen), um die Dokumente zu analysieren. Amazon Comprehend speichert die Ergebnisse der Analyse in dem S3-Bucket/-Ordner, den Sie in der Anforderung angeben.

## Themen

- [Einzeldokumentverarbeitung](#)
- [Synchrone Verarbeitung mehrerer Dokumente](#)
- [Asynchrone Batchverarbeitung](#)

## Einzeldokumentverarbeitung

Einzeldokumentoperationen sind synchrone Operationen, die die Ergebnisse der Dokumentanalyse direkt an Ihre Anwendung zurückgeben. Verwenden Sie synchrone Einzeldokumentoperationen, wenn Sie eine interaktive Anwendung erstellen, die jeweils für ein Dokument funktioniert.

Weitere Informationen zu den synchronen API-Operationen finden Sie unter [Echtzeitanalyse mit den integrierten Modellen](#) (für die Konsole) und [Echtzeitanalyse mit der API](#).

## Synchrone Verarbeitung mehrerer Dokumente

Wenn Sie mehrere Dokumente verarbeiten möchten, können Sie die Batch\* API-Operationen verwenden, um mehr als ein Dokument gleichzeitig an Amazon Comprehend zu senden. Sie können bis zu 25 Dokumente in jeder Anfrage senden. Amazon Comprehend sendet eine Liste von Antworten zurück, eine für jedes Dokument in der Anforderung. Anforderungen, die mit diesen

Operationen gestellt werden, sind synchron. Ihre Anwendung ruft die -Operation auf und wartet dann auf die Antwort vom Service.

Die Verwendung der -Batch\*Operationen ist identisch mit dem Aufrufen der einzelnen Dokument-APIs für jedes der Dokumente in der Anforderung. Die Verwendung dieser APIs kann zu einer besseren Leistung für Ihre Anwendungen führen.

Die Eingabe für jede der APIs ist eine JSON-Struktur, die die zu verarbeitenden Dokumente enthält. Für alle Operationen außer müssen BatchDetectDominantLanguageSie die Eingabesprache festlegen. Sie können für jede Anfrage nur eine Eingabesprache festlegen. Im Folgenden finden Sie beispielsweise die Eingabe für die -BatchDetectEntitiesOperation. Es enthält zwei Dokumente und ist in englischer Sprache.

```
{
  "LanguageCode": "en",
  "TextList": [
    "I have been living in Seattle for almost 4 years",
    "It is raining today in Seattle"
  ]
}
```

Die Antwort einer -Batch\*Operation enthält zwei Listen, die ResultList und die ErrorList. Die ResultList enthält einen Datensatz für jedes Dokument, das erfolgreich verarbeitet wurde. Das Ergebnis für jedes Dokument in der Anforderung ist identisch mit dem Ergebnis, das Sie erhalten würden, wenn Sie eine einzelne Dokumentoperation für das Dokument ausführen würden. Den Ergebnissen für jedes Dokument wird ein Index zugewiesen, der auf der Reihenfolge der Dokumente in der Eingabedatei basiert. Die Antwort des BatchDetectEntities Vorgangs lautet:

```
{
  "ResultList" : [
    {
      "Index": 0,
      "Entities": [
        {
          "Text": "Seattle",
          "Score": 0.95,
          "Type": "LOCATION",
          "BeginOffset": 22,
          "EndOffset": 29
        }
      ],
    },
  ],
}
```

```

    {
      "Text": "almost 4 years",
      "Score": 0.89,
      "Type": "QUANTITY",
      "BeginOffset": 34,
      "EndOffset": 48
    }
  ]
},
{
  "Index": 1,
  "Entities": [
    {
      "Text": "today",
      "Score": 0.87,
      "Type": "DATE",
      "BeginOffset": 14,
      "EndOffset": 19
    },
    {
      "Text": "Seattle",
      "Score": 0.96,
      "Type": "LOCATION",
      "BeginOffset": 23,
      "EndOffset": 30
    }
  ]
}
],
"ErrorList": []
}

```

Wenn in der Anforderung ein Fehler auftritt, enthält die Antwort einen `ErrorList`, der die Dokumente identifiziert, die einen Fehler enthielten. Das Dokument wird durch seinen Index in der Eingabeliste identifiziert. Die folgende Eingabe für die `-BatchDetectLanguageOperation` enthält beispielsweise ein Dokument, das nicht verarbeitet werden kann:

```

{
  "TextList": [
    "hello friend",
    "$$$$$$",
    "hola amigo"
  ]
}

```

```
}
```

Die Antwort von Amazon Comprehend enthält eine Fehlerliste, die das Dokument identifiziert, das einen Fehler enthielt:

```
{
  "ResultList": [
    {
      "Index": 0,
      "Languages": [
        {
          "LanguageCode": "en",
          "Score": 0.99
        }
      ]
    },
    {
      "Index": 2,
      "Languages": [
        {
          "LanguageCode": "es",
          "Score": 0.82
        }
      ]
    }
  ],
  "ErrorList": [
    {
      "Index": 1,
      "ErrorCode": "InternalServerError",
      "ErrorMessage": "Unexpected Server Error. Please try again."
    }
  ]
}
```

Weitere Informationen zu den synchronen Batch-API-Operationen finden Sie unter [Batch-APIs in Echtzeit](#).

## Asynchrone Batchverarbeitung

Um große Dokumente und große Sammlungen von Dokumenten zu analysieren, verwenden Sie die asynchronen Operationen von Amazon Comprehend.

Um eine Sammlung von Dokumenten zu analysieren, führen Sie in der Regel die folgenden Schritte aus:

1. Speichern Sie die Dokumente in einem Amazon S3-Bucket.
2. Starten Sie einen oder mehrere Analyseaufträge, um die Dokumente zu analysieren.
3. Überwachen Sie den Fortschritt der Analyseaufträge.
4. Rufen Sie die Ergebnisse der Analyse aus einem S3-Bucket ab, wenn der Auftrag abgeschlossen ist.

Weitere Informationen zur Verwendung der asynchronen API-Operationen finden Sie unter [Ausführen von Analyseaufträgen mit der Konsole](#) (Konsole) und [Asynchrone Analyseaufträge mit der API](#).

# In Amazon Comprehend unterstützte Sprachen

Amazon Comprehend unterstützt eine Vielzahl von Sprachen für seine verschiedenen Funktionen. Die unterstützten Sprachen und die Funktionen, die sie unterstützen, finden Sie in den folgenden Tabellen.

## Themen


- [Unterstützte Sprachen](#)
- [Sprachen, die von Amazon Comprehend-Funktionen unterstützt werden](#)

## Unterstützte Sprachen

Amazon Comprehend (mit Ausnahme der Funktion zur Erkennung dominanter Sprachen) unterstützt die folgenden Sprachen für eine oder mehrere Funktionen.

Code	Sprache
de	Deutsch
en	Englisch
es	Spanisch
it	Italienisch
pt	Portugiesisch
fr	Französisch
ja	Japanisch
ko	Koreanisch
hi	Hindi
ar	Arabisch
zh	Chinesisch (vereinfacht)

Code	Sprache
zh-TW	Chinesisch (traditionell)

 Note

Amazon Comprehend identifiziert die Sprache anhand von Kennungen aus RFC 5646 – wenn es eine zweistellige ISO 639-1-Kennung mit einem regionalen Untertag gibt./ Bei Bedarf wird dies verwendet. Andernfalls wird der Code ISO 639-2 aus 3 Buchstaben verwendet.

Weitere Informationen zu RFC 5646 finden Sie unter [Tags zur Identifizierung von Sprachen](#) auf der Website der IETF-Tools.

## Sprachen, die von Amazon Comprehend-Funktionen unterstützt werden

Funktion	Unterstützte Sprachen
<a href="#">Dominante Sprache</a>	Siehe <a href="#">Dominante Sprache</a> .
<a href="#">Entitäten</a>	Alle unterstützten Sprachen.
<a href="#">Schlüsselphrasen</a>	Alle unterstützten Sprachen.
<a href="#">Erkennung von PII-Entitäten</a>	Englisch und Spanisch.
<a href="#">Kennzeichnung von PII-Entitäten</a>	Englisch und Spanisch.
<a href="#">Stimmung</a>	Alle unterstützten Sprachen.
<a href="#">Gezielte Stimmung</a>	Englisch.
<a href="#">Syntaxanalyse</a>	Deutsch (de), Englisch (en), Spanisch (es), Französisch (fr), Italienisch (it) und Portugiesisch (pt).

Funktion	Unterstützte Sprachen
<a href="#">Themenmodellierung</a>	Nicht von der verwendeten Sprache abhängig. Unterstützt keine zeichenbasierten Sprachen wie Chinesisch, Japanisch und Koreanisch.
<a href="#">Benutzerdefinierte Klassifizierung</a>	Klartextmodelle unterstützen die folgenden Sprachen: Deutsch (de), Englisch (en), Spanisch (es), Französisch (fr), Italienisch (it) und Portugiesisch (pt).  <a href="#">Native Dokumentmodelle</a> unterstützen nur englische Dokumente.
<a href="#">Erkennung benutzerdefinierter Entitäten</a>	Deutsch (de), Englisch (en), Spanisch (es), Französisch (fr), Italienisch (it) und Portugiesisch (pt).  Benutzerdefinierte Entitätserkennung für PDF und Word unterstützt nur englische Dokumente.



# Einrichtung

Bevor Sie Amazon Comprehend zum ersten Mal verwenden, führen Sie die folgenden Aufgaben aus.

Aufgaben einrichten

- [Melde dich an für ein AWS-Konto](#)
- [Erstellen Sie einen Benutzer mit Administratorzugriff](#)
- [Richten Sie das AWS Command Line Interface \(AWS CLI\) ein](#)
- [Erteilen programmgesteuerten Zugriffs](#)

## Melde dich an für ein AWS-Konto

Wenn Sie noch keine haben AWS-Konto, führen Sie die folgenden Schritte aus, um eine zu erstellen.

Um sich für eine anzumelden AWS-Konto

1. Öffnen Sie <https://portal.aws.amazon.com/billing/signup>.
2. Folgen Sie den Online-Anweisungen.

Bei der Anmeldung müssen Sie auch einen Telefonanruf entgegennehmen und einen Verifizierungscode über die Telefontasten eingeben.

Wenn Sie sich für eine anmelden AWS-Konto, Root-Benutzer des AWS-Kontos wird eine erstellt. Der Root-Benutzer hat Zugriff auf alle AWS-Services und Ressourcen des Kontos. Aus Sicherheitsgründen sollten Sie einem Benutzer Administratorzugriff zuweisen und nur den Root-Benutzer verwenden, um [Aufgaben auszuführen, für die Root-Benutzerzugriff erforderlich](#) ist.

AWS sendet Ihnen nach Abschluss des Anmeldevorgangs eine Bestätigungs-E-Mail. Sie können jederzeit Ihre aktuelle Kontoaktivität anzeigen und Ihr Konto verwalten. Rufen Sie dazu <https://aws.amazon.com/> auf und klicken Sie auf Mein Konto.

## Erstellen Sie einen Benutzer mit Administratorzugriff

Nachdem Sie sich für einen angemeldet haben AWS-Konto, sichern Sie Ihren Root-Benutzer des AWS-Kontos AWS IAM Identity Center, aktivieren und erstellen Sie einen Administratorbenutzer, sodass Sie den Root-Benutzer nicht für alltägliche Aufgaben verwenden.

## Sichern Sie Ihre Root-Benutzer des AWS-Kontos

1. Melden Sie sich [AWS Management Console](#) als Kontoinhaber an, indem Sie Root-Benutzer auswählen und Ihre AWS-Konto E-Mail-Adresse eingeben. Geben Sie auf der nächsten Seite Ihr Passwort ein.

Hilfe bei der Anmeldung mit dem Root-Benutzer finden Sie unter [Anmelden als Root-Benutzer](#) im AWS-Anmeldung Benutzerhandbuch zu.

2. Aktivieren Sie die Multi-Faktor-Authentifizierung (MFA) für den Root-Benutzer.

Anweisungen finden Sie unter [Aktivieren eines virtuellen MFA-Geräts für Ihren AWS-Konto Root-Benutzer \(Konsole\)](#) im IAM-Benutzerhandbuch.

## Erstellen Sie einen Benutzer mit Administratorzugriff

1. Aktivieren Sie das IAM Identity Center.

Anweisungen finden Sie unter [Aktivieren AWS IAM Identity Center](#) im AWS IAM Identity Center Benutzerhandbuch.

2. Gewähren Sie einem Benutzer in IAM Identity Center Administratorzugriff.

Ein Tutorial zur Verwendung von IAM-Identity-Center-Verzeichnis als Identitätsquelle finden [Sie unter Benutzerzugriff mit der Standardeinstellung konfigurieren IAM-Identity-Center-Verzeichnis](#) im AWS IAM Identity Center Benutzerhandbuch.

## Melden Sie sich als Benutzer mit Administratorzugriff an

- Um sich mit Ihrem IAM-Identity-Center-Benutzer anzumelden, verwenden Sie die Anmelde-URL, die an Ihre E-Mail-Adresse gesendet wurde, als Sie den IAM-Identity-Center-Benutzer erstellt haben.

Hilfe bei der Anmeldung mit einem IAM Identity Center-Benutzer finden Sie [im AWS-Anmeldung Benutzerhandbuch unter Anmeldung beim AWS Zugriffsportal](#).

## Weisen Sie weiteren Benutzern Zugriff zu

1. Erstellen Sie in IAM Identity Center einen Berechtigungssatz, der der bewährten Methode zur Anwendung von Berechtigungen mit den geringsten Rechten folgt.

Anweisungen finden Sie im Benutzerhandbuch unter [Einen Berechtigungssatz erstellen](#).AWS IAM Identity Center

2. Weisen Sie Benutzer einer Gruppe zu und weisen Sie der Gruppe dann Single Sign-On-Zugriff zu.

Anweisungen finden [Sie im AWS IAM Identity Center Benutzerhandbuch unter Gruppen hinzufügen](#).

## Richten Sie das AWS Command Line Interface (AWS CLI) ein

Sie benötigen das nicht AWS CLI , um die Schritte in den Übungen Erste Schritte auszuführen. Für einige andere Übungen in diesem Handbuch wird sie jedoch benötigt. Wenn Sie möchten, können Sie diesen Schritt überspringen und zu [Erste Schritte mit Amazon Comprehend](#) einem AWS CLI späteren Zeitpunkt wechseln und ihn einrichten.

Um das zu installieren und zu konfigurieren AWS CLI

1. Installieren Sie das AWS CLI. Anweisungen finden Sie unter dem folgenden Thema im AWS Command Line Interface Benutzerhandbuch:

[Installation oder Aktualisierung der neuesten Version von AWS Command Line Interface](#)

2. Konfigurieren Sie AWS CLI. Anweisungen finden Sie unter dem folgenden Thema im AWS Command Line Interface Benutzerhandbuch:

[Konfigurieren von AWS Command Line Interface](#)

## Erteilen programmgesteuerten Zugriffs

Benutzer benötigen programmatischen Zugriff, wenn sie mit AWS außerhalb des AWS Management Console interagieren möchten. Die Art und Weise, wie programmatischer Zugriff gewährt wird, hängt vom Benutzertyp ab, der zugreift. AWS

Um Benutzern programmgesteuerten Zugriff zu gewähren, wählen Sie eine der folgenden Optionen.

Welcher Benutzer benötigt programmgesteuerten Zugriff?	Bis	Von
<p>Mitarbeiteridentität</p> <p>(Benutzer, die in IAM Identity Center verwaltet werden)</p>	<p>Verwenden Sie temporäre Anmeldeinformationen, um programmatische Anfragen an die AWS CLI, AWS SDKs oder APIs zu signieren. AWS</p>	<p>Befolgen Sie die Anweisungen für die Schnittstelle, die Sie verwenden möchten.</p> <ul style="list-style-type: none"> <li>• Informationen zu den AWS CLI finden Sie unter <a href="#">Konfiguration der AWS CLI zu AWS IAM Identity Center verwenden</a> im AWS Command Line Interface Benutzerhandbuch.</li> <li>• Informationen zu AWS SDKs, Tools und AWS APIs finden Sie unter <a href="#">IAM Identity Center-Authentifizierung im Referenzhandbuch</a> für AWS SDKs und Tools.</li> </ul>
IAM	<p>Verwenden Sie temporäre Anmeldeinformationen, um programmatische Anfragen an die AWS CLI, AWS SDKs oder APIs zu signieren. AWS</p>	<p>Folgen Sie den Anweisungen unter <a href="#">Verwenden temporärer Anmeldeinformationen mit AWS Ressourcen</a> im IAM-Benutzerhandbuch.</p>
IAM	<p>(Nicht empfohlen)</p> <p>Verwenden Sie langfristige Anmeldeinformationen, um programmatische Anfragen an die AWS CLI, AWS SDKs oder APIs zu signieren. AWS</p>	<p>Befolgen Sie die Anweisungen für die Schnittstelle, die Sie verwenden möchten.</p> <ul style="list-style-type: none"> <li>• Informationen dazu finden Sie unter <a href="#">Authentifizierung mithilfe von IAM-Benutzeranmeldedaten im Benutzerhandbuch</a>. AWS</li> </ul>

Welcher Benutzer benötigt programmgesteuerten Zugriff?	Bis	Von
		<p>CLIAWS Command Line Interface</p> <ul style="list-style-type: none"><li>• Informationen zu AWS SDKs und Tools finden Sie unter <a href="#">Authentifizieren mit langfristigen Anmeldeinformationen</a> im Referenzhandbuch für AWS SDKs und Tools.</li><li>• Informationen zu AWS APIs finden Sie unter <a href="#">Verwaltung von Zugriffsschlüsseln für IAM-Benutzer</a> im IAM-Benutzerhandbuch.</li></ul>

# Erste Schritte mit Amazon Comprehend

In der folgenden Übung wird die Amazon Comprehend-Konsole verwendet, um einen asynchronen Entitätserkennungsauftrag zu erstellen und auszuführen. Bei dieser Übung wird davon ausgegangen, dass Sie mit Amazon Simple Storage Service (Amazon S3) vertraut sind. Ein einfacheres Beispiel finden Sie unter [Echtzeitanalyse mit den integrierten Modellen](#).

So erstellen Sie einen Auftrag zur Entitätserkennung

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon Comprehend-Konsole unter <https://console.aws.amazon.com/comprehend/>
2. Wählen Sie im linken Menü Analyseaufträge und dann Auftrag erstellen aus.
3. Geben Sie unter Auftragseinstellungen dem Auftrag einen Namen. Der Name muss innerhalb der Region und des Kontos eindeutig sein.
4. Wählen Sie für Analysetyp die Option Entitäten aus.
5. Wählen Sie für Sprache die Sprache der Eingabedokumente aus.
6. Wählen Sie unter Eingabedaten für Datenquelle die Option Beispieldokumente aus. Die Konsole legt den S3-Speicherort als Ordner fest, der die öffentlichen Beispiele enthält.
7. Fügen Sie unter Ausgabedaten am S3-Speicherort die URL oder den Ordnerspeicherort in Amazon S3 für die Ausgabedateien ein.
8. Wählen Sie im Abschnitt Zugriffsberechtigungen die Option Erstellen einer IAM-Rolle aus. Die Konsole erstellt eine neue IAM-Rolle mit den richtigen Berechtigungen für Amazon Comprehend für den Zugriff auf die Eingabe- und Ausgabe-Buckets.
9. Wenn Sie das Formular ausgefüllt haben, wählen Sie Auftrag erstellen, um den Themenerkennungsauftrag zu erstellen und zu starten.

Der neue Auftrag wird in der Auftragsliste angezeigt, wobei das Statusfeld den Status des Auftrags anzeigt. Das Feld kann IN\_PROGRESS für einen Auftrag sein, der gerade verarbeitet wird, COMPLETED für einen Auftrag, der erfolgreich abgeschlossen wurde, und FAILED für einen Auftrag, der einen Fehler aufweist.

10. Wählen Sie den Auftrag aus, um den Bereich Auftragsdetails zu öffnen.
11. Wählen Sie unter Ausgabe unter Speicherort der Ausgabedaten den Link aus, um die Amazon S3-Konsole zu öffnen.
12. Wählen Sie in der Amazon S3-Konsole Herunterladen und speichern Sie die `output.tar.gz` Datei.

13. Dekomprimieren Sie die Datei und speichern Sie sie als Json-Datei.
14. [the section called “Entitäten”](#) Eine Beschreibung der Entitätstypen und der Felder für jede erkannte Entität finden Sie unter .

# Analyse mit der Amazon Comprehend-Konsole

Sie können die Amazon Comprehend-Konsole verwenden, um Dokumente in Echtzeit zu analysieren oder asynchrone Analyseaufträge auszuführen.

Mithilfe der Echtzeitanalyse mit integrierten Modellen können Sie Entitäten erkennen, Schlüsselphrasen extrahieren, Primärsprache erkennen, PII erkennen, Stimmungen ermitteln, gezielte Stimmungen analysieren und die Syntax analysieren.

Sie können Analyseaufträge mit den integrierten Modellen ausführen, um Erkenntnisse wie Entitäten, Ereignisse, Wortgruppen, Primärsprache, Stimmung, gezielte Stimmung und persönlich identifizierbare Informationen (PII) zu finden. Sie können auch Themenmodellierungsaufträge ausführen.

Die Konsole unterstützt auch Echtzeit- und asynchrone Analysen mit benutzerdefinierten Modellen. Weitere Informationen finden Sie unter [Benutzerdefinierte Klassifizierung](#) und [Erkennung benutzerdefinierter Entitäten](#).

Themen

- [Echtzeitanalyse mit den integrierten Modellen](#)
- [Ausführen von Analyseaufträgen mit der Konsole](#)

## Echtzeitanalyse mit den integrierten Modellen

Sie können die Amazon Comprehend Comprehend-Konsole verwenden, um eine Echtzeitanalyse eines UTF-8-codierten Textdokuments durchzuführen. Das Dokument kann Englisch oder eine der anderen von Amazon Comprehend unterstützten Sprachen sein. Die Ergebnisse werden in der Konsole angezeigt, sodass Sie die Analyse überprüfen können.

Um mit der Analyse von Dokumenten zu beginnen, melden Sie sich bei der [Amazon Comprehend Comprehend-Konsole](#) an AWS Management Console und öffnen Sie sie.

Sie können den Beispieltext durch Ihren eigenen Text ersetzen und dann Analysieren wählen, um eine Analyse Ihres Textes zu erhalten. Unter dem zu analysierenden Text werden im Ergebnisbereich weitere Informationen zum Text angezeigt.



Führen Sie mithilfe des integrierten Modells eine Echtzeitanalyse durch

1. [Melden Sie sich bei der Amazon Comprehend Comprehend-Konsole an AWS Management Console und öffnen Sie sie unter https://console.aws.amazon.com/comprehend/](https://console.aws.amazon.com/comprehend/)
2. Wählen Sie im linken Menü die Option Echtzeitanalyse aus.
3. Wählen Sie unter Eingabetyp die Option Integriert als Analysetyp aus.
4. Geben Sie den Text ein, den Sie analysieren möchten.
5. Wählen Sie Analysieren. Die Konsole zeigt die Ergebnisse der Textanalyse im Bereich Einblicke an. Das Insights-Panel enthält eine Registerkarte für jeden Erkenntnistyp. In den folgenden Abschnitten werden die Ergebnisse für den Insight-Typ beschrieben.

Themen

- [Entitäten](#)
- [Wichtige Ausdrücke](#)
- [Sprache](#)
- [Persönlich Identifizierbare Informationen \(PII\)](#)
- [Stimmung](#)
- [Gezielte Stimmung](#)
- [Syntax](#)

## Entitäten

Auf der Registerkarte Entitäten werden alle Entitäten, ihre Kategorie und das Vertrauensniveau aufgeführt, das Amazon Comprehend im Eingabetext erkannt hat. Die Ergebnisse sind farblich gekennzeichnet, um auf verschiedene Entitätstypen wie Organisationen, Standorte, Daten und Personen hinzuweisen. Weitere Informationen finden Sie unter [Entitäten](#).

**Insights** [Info](#)

---

Entities
Key phrases
Language
PII
Sentiment
Targeted sentiment
Syntax

**Analyzed text**

Hello [Zhang Wei](#), I am [John](#). Your [AnyCompany Financial Services, LLC](#) credit card account [1111-0000-1111-0008](#) has a minimum payment of [\\$24.53](#) that is due by [July 31st](#). Based on your autopay settings, we will withdraw your payment on the due date from your bank account number [XXXXXX1111](#) with the routing number [XXXXX0000](#).

Customer feedback for [Sunshine Spa](#), [123 Main St](#), Anywhere. Send comments to [Alice](#) at [sunspa@mail.com](mailto:sunspa@mail.com).

I enjoyed visiting the spa. It was very comfortable but it was also very expensive. The amenities were ok but the service made the spa a great experience.

**▼ Results**

< 1 2 > ⚙️

Entity	Type	Confidence
Zhang Wei	Person	0.99+
John	Person	0.99+
AnyCompany Financial Services, LLC	Organization	0.99+
1111-0000-1111-0008	Other	0.99+
\$24.53	Quantity	0.99+
July 31st	Date	0.99+
XXXXXX1111	Other	0.98
XXXXX0000	Other	0.96
Sunshine Spa	Organization	0.98
123 Main St	Location	0.98

**▶ Application integration**

## Wichtige Ausdrücke

Auf der Registerkarte Schlüsselbegriffe werden wichtige Nominalphrasen, die Amazon Comprehend im Eingabetext erkannt hat, sowie das zugehörige Konfidenzniveau aufgeführt. Weitere Informationen finden Sie unter [Schlüsselphrasen](#).

## Insights Info

---

Entities
Key phrases
Language
PII
Sentiment
Targeted sentiment
Syntax

**Analyzed text**

Hello [Zhang Wei](#), I am [John](#). [Your AnyCompany Financial Services, LLC credit card account 1111-0000-1111-0008](#) has a [minimum payment of \\$24.53](#) that is due by [July 31st](#). Based on [your autopay settings](#), we will withdraw [your payment on the due date](#) from [your bank account number XXXXXX1111](#) with [the routing number XXXXX0000](#).

[Customer feedback for Sunshine Spa, 123 Main St, Anywhere](#). Send [comments to Alice at sunspa@mail.com](#).

I enjoyed visiting [the spa](#). It was very comfortable but it was also very expensive. [The amenities](#) were ok but [the service made the spa a great experience](#).

**▼ Results**

< 1 2 3 > ⚙️

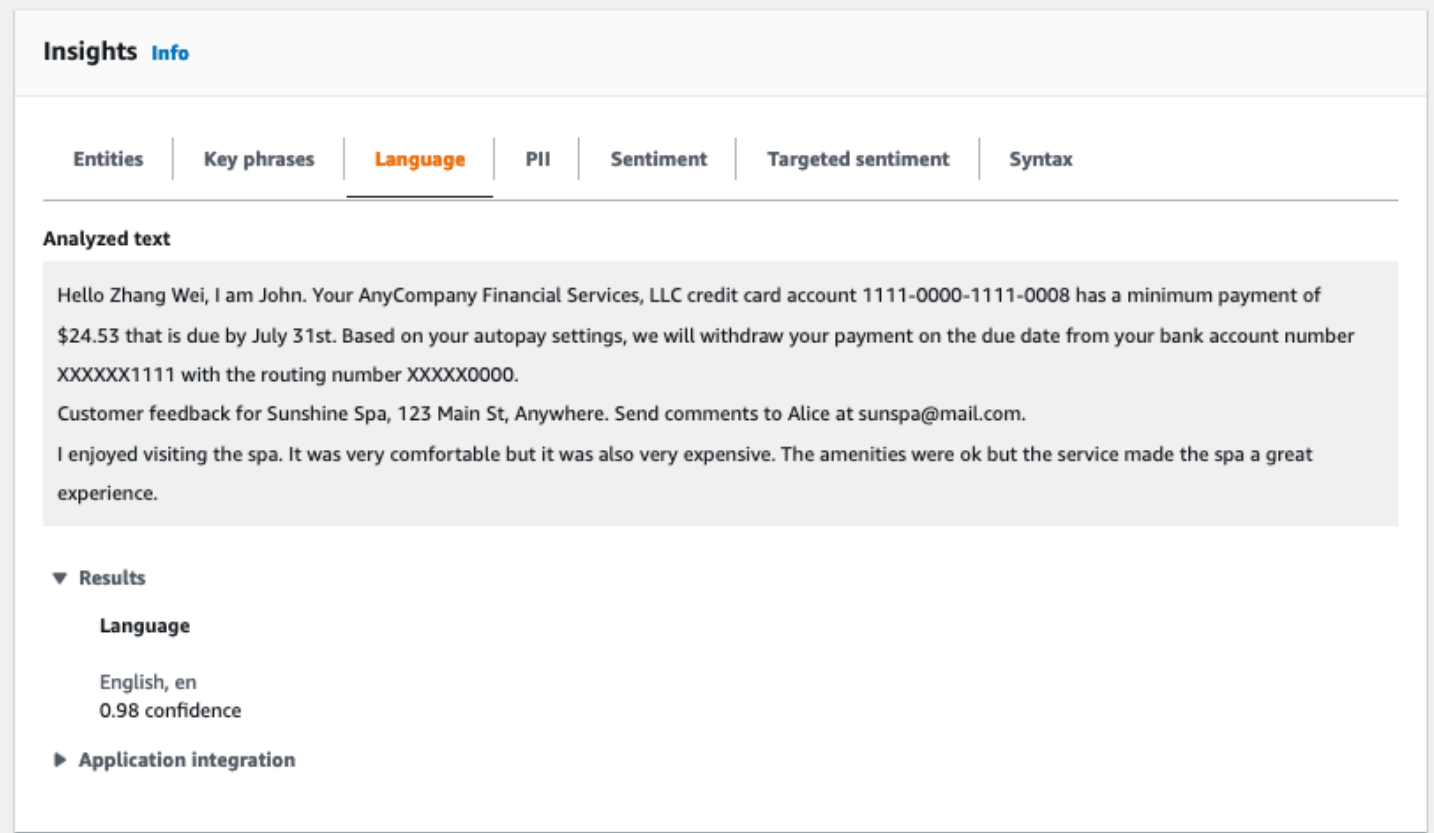
Key phrases	Confidence
Zhang Wei	0.93
John	0.99+
Your AnyCompany Financial Services	0.98
LLC credit card account 1111-0000-1111-0008	0.87
a minimum payment	0.99+
\$24.53	0.99+
July 31st	0.99+
your autopay settings	0.99+
your payment	0.99+
the due date	0.99+

**▶ Application integration**

## Sprache

Auf der Registerkarte Sprache werden die dominante Sprache des Textes und die Vertrauensstufe von Amazon Comprehend angezeigt, dass die dominante Sprache korrekt erkannt wurde. Amazon

Comprehend kann 100 Sprachen erkennen. Weitere Informationen finden Sie unter [Dominante Sprache](#).



The screenshot displays the 'Insights Info' section of the Amazon Comprehend interface. It features a navigation bar with tabs for 'Entities', 'Key phrases', 'Language', 'PII', 'Sentiment', 'Targeted sentiment', and 'Syntax'. The 'Language' tab is selected and highlighted in orange. Below the navigation bar, the 'Analyzed text' section contains three paragraphs of sample text. The 'Results' section is expanded to show 'Language' analysis, which identifies the text as 'English, en' with a '0.98 confidence' score. There is also a link for 'Application integration'.

**Insights Info**

Entities | Key phrases | **Language** | PII | Sentiment | Targeted sentiment | Syntax

**Analyzed text**

Hello Zhang Wei, I am John. Your AnyCompany Financial Services, LLC credit card account 1111-0000-1111-0008 has a minimum payment of \$24.53 that is due by July 31st. Based on your autopay settings, we will withdraw your payment on the due date from your bank account number XXXXXX1111 with the routing number XXXXX0000.

Customer feedback for Sunshine Spa, 123 Main St, Anywhere. Send comments to Alice at sunspa@mail.com.

I enjoyed visiting the spa. It was very comfortable but it was also very expensive. The amenities were ok but the service made the spa a great experience.

▼ **Results**

**Language**

English, en  
0.98 confidence

► **Application integration**

## Persönlich Identifizierbare Informationen (PII)

Auf der Registerkarte „PII“ werden Entitäten in Ihrem Eingabetext aufgeführt, die personenbezogene Daten (PII) enthalten. Eine PII-Entität ist ein Textverweis auf personenbezogene Daten, die zur Identifizierung einer Person verwendet werden könnten, z. B. eine Adresse, Bankkontonummer oder Telefonnummer. Weitere Informationen finden Sie unter [Erkennung von PII-Entitäten](#).

Die Registerkarte PII bietet zwei Analysemodi:

- Offsets
- Labels

### Offsets

Der Analysemodus Offsets identifiziert die Position personenbezogener Daten in Ihren Textdokumenten. Weitere Informationen finden Sie unter [Suchen Sie nach PII-Entitäten](#).

**Insights** [Info](#)

---

Entities
Key phrases
Language
PII
Sentiment
Targeted sentiment
Syntax

Personally identifiable information (PII) analysis mode

**Offsets**  
Identify the location of PII in your text documents.

**Labels**  
Label text documents with PII.

**Analyzed text**

Hello [Zhang Wei](#), I am [John](#). Your AnyCompany Financial Services, LLC credit card account [1111-0000-1111-0008](#) has a minimum payment of \$24.53 that is due by [July 31st](#). Based on your autopay settings, we will withdraw your payment on the due date from your bank account number [XXXXXX1111](#) with the routing number [XXXXX0000](#).

Customer feedback for Sunshine Spa, [123 Main St](#), Anywhere. Send comments to [Alice](#) at [sunspa@mail.com](mailto:sunspa@mail.com).

I enjoyed visiting the spa. It was very comfortable but it was also very expensive. The amenities were ok but the service made the spa a great experience.

▼ **Results**

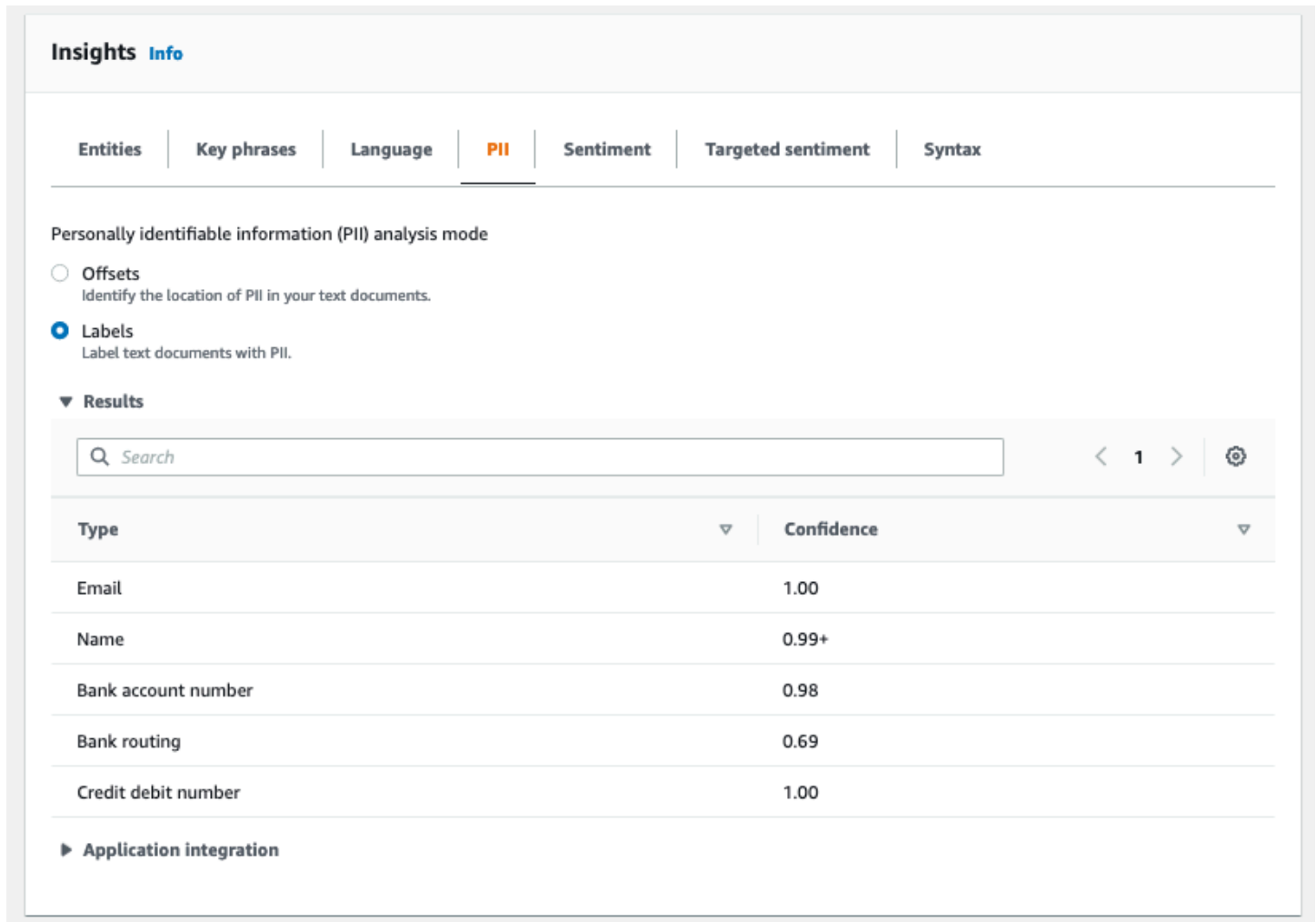
< 1 > ⚙️

Entity	Type	Confidence
Zhang Wei	Name	0.99+
John	Name	0.99+
1111-0000-1111-0008	Credit debit number	0.99+
July 31st	Date time	0.99+
XXXXXX1111	Bank account number	0.99+
XXXXX0000	Bank routing	0.99+
123 Main St	Address	0.99+
Alice	Name	0.99+
sunspa@mail.com	Email	0.99+

► **Application integration**

## Labels

Der Analysemodus Labels prüft, ob in Ihrem Textdokument personenbezogene Daten vorhanden sind, und gibt die Labels der identifizierten PII-Entitätstypen zurück. Weitere Informationen finden Sie unter [Kennzeichnung von PII-Entitäten](#).



The screenshot displays the 'Insights Info' section of the Amazon Comprehend console. The 'PII' tab is selected, showing 'Personally identifiable information (PII) analysis mode'. The 'Labels' option is chosen, indicating that text documents are labeled with PII. Below this, a 'Results' section shows a search bar and a table of identified PII entities. The table lists five entity types with their respective confidence scores: Email (1.00), Name (0.99+), Bank account number (0.98), Bank routing (0.69), and Credit debit number (1.00). At the bottom, there is a link for 'Application integration'.

Type	Confidence
Email	1.00
Name	0.99+
Bank account number	0.98
Bank routing	0.69
Credit debit number	1.00

## Stimmung

Auf der Registerkarte Stimmung wird die vorherrschende Stimmung des Textes angezeigt. Die Stimmung kann als neutral, positiv, negativ oder gemischt bewertet werden. In diesem Fall hat jede Stimmung eine Vertrauensbewertung, die eine Schätzung von Amazon Comprehend dafür liefert, dass diese Stimmung vorherrscht. Weitere Informationen finden Sie unter [Stimmung](#).

The screenshot displays the 'Insights Info' section of the Amazon Comprehend console. It features a navigation bar with tabs for 'Entities', 'Key phrases', 'Language', 'PII', 'Sentiment' (which is selected and highlighted in orange), 'Targeted sentiment', and 'Syntax'. Below the navigation bar, the 'Analyzed text' section contains three paragraphs of sample text. The first paragraph is a credit card statement, the second is a customer feedback message, and the third is a personal review. Underneath, the 'Results' section shows a 'Sentiment' breakdown with four categories: Neutral (0.56 confidence), Positive (0.10 confidence), Negative (0.19 confidence), and Mixed (0.14 confidence). At the bottom, there is a link for 'Application integration'.

## Gezielte Stimmung

Eine gezielte Stimmungsanalyse identifiziert die Stimmungen, die gegenüber den im Text genannten Unternehmen geäußert wurden. Amazon Comprehend weist jeder Erwähnung einer Entität eine Stimmungsbewertung sowie eine Vertrauensbewertung und andere Informationen zu. Eine Stimmungsbewertung kann neutral, positiv, negativ oder gemischt sein.

Im Bereich Analysierter Text unterstreicht die Konsole jede der analysierten Entitäten. Die Farbe des unterstrichenen Textes gibt die allgemeine Stimmung der Entität an. Wenn Sie den Mauszeiger über eine Entität bewegen, zeigt die Konsole zusätzliche Informationen in einem Popup-Fenster an.

**Insights** [Info](#)

Entities | Key phrases | Language | PII | Sentiment | **Targeted sentiment** | Syntax

**Analyzed text** 
■ Positive 
 ■ Neutral 
 ■ Negative 
 ■ Mixed

Hello Zhang Wei, I am John. Your AnyCompany Financial Services, LLC credit card account 1111-0000-1111-0008 has a minimum payment of \$100.00 per month. 31st. Based on your autopay settings, we will withdraw your payment on the due date from your credit card account ending in X1111 with the routing number XXXXX0000.

Spa, 123 Main St, Anywhere. Send comments to Alice at sunspa@mail.com.

The spa was very comfortable but it was also very expensive. The amenities were ok but the service made the spa a great place to visit.

Entity type: PERSON ×

Entity confidence: 0.99+

Sentiment: NEUTRAL

Sentiment confidence: 0.99+

Total related entities: 5

Die Ergebnistabelle enthält zusätzliche Details zu jeder Entität. Wenn es mehrere Erwähnungen derselben Entität gibt, die als Koreferenzgruppe bezeichnet wird, werden diese Erwähnungen in der Tabelle als zusammenklappbare Gruppe von Zeilen angezeigt, die der Hauptentität zugeordnet sind.

Im folgenden Beispiel handelt es sich bei der Entität um eine Person namens Zhang Wei. Bei der gezielten Stimmungsanalyse wird erkannt, dass sich jede Erwähnung Ihrer Person auf dieselbe Person bezieht. In der Konsole werden diese Erwähnungen als Untereinträge der Hauptentität angezeigt.



## Analyzed text

■ Positive
 ■ Neutral
 ■ Negative
 ■ Mixed

Hello Zhang Wei , I am John . Your AnyCompany Financial Services, LLC credit card account 1111-0000-1111-0008 has a minimum payment of \$24.53 that is due by July 31st . Based on your autopay settings, we will withdraw your payment on the due date from your bank account number XXXXXX1111 with the routing number XXXXX0000.  
 Customer feedback for Sunshine Spa , 123 Main St , Anywhere . Send comments to Alice at sunspa@mail.com .  
 I enjoyed visiting the spa . It was very comfortable but it was also very expensive. The amenities were ok but the service made the spa a great experience.

## ▼ Results

Entity	Entity type	Entity score	Primary sentiment	Positive score	Ne
☐ Zhang Wei (5)	PERSON	-	— NEUTRAL	-	-
— your	PERSON	0.99+	— NEUTRAL	0	0
— your	PERSON	0.67	— NEUTRAL	0	0
— Your	ORGANIZATION	0.94	— NEUTRAL	0	0
— your	PERSON	0.99+	— NEUTRAL	0	0
— Zhang Wei	PERSON	0.99+	— NEUTRAL	0.00	0

Wenn der Text, den Sie analysieren, keine gezielte Stimmungslage enthält [Entitätstypen](#), zeigt die gezielte Stimmungsanalyse ein leeres Ergebnisfeld an.

Weitere Informationen zur Verwendung der Konsole für gezielte Stimmungsanalysen in Echtzeit finden Sie unter [Echtzeitanalyse mit der Konsole](#)

## Syntax

Die Registerkarte Syntax zeigt eine Aufschlüsselung der einzelnen Textelemente, ihrer Wortart und des zugehörigen Konfidenzwerts. Weitere Informationen finden Sie unter [Syntaxanalyse](#).

**Insights** [Info](#)

---

Entities
Key phrases
Language
PII
Sentiment
Targeted sentiment
Syntax

**Analyzed text**

Hello Zhang Wei, I am John. Your AnyCompany Financial Services, LLC credit card account 1111-0000-1111-0008 has a minimum payment of \$ 24.53 that is due by July 31st. Based on your autopay settings, we will withdraw your payment on the due date from your bank account number XXXXXX1111 with the routing number XXXXX0000.

Customer feedback for Sunshine Spa, 123 Main St, Anywhere. Send comments to Alice at sunspa@mail.com.

I enjoyed visiting the spa. It was very comfortable but it was also very expensive. The amenities were ok but the service made the spa a great experience.

**▼ Results**

< 1 2 3 4 5 6 7 ... 11 > ⚙

Word	Part of speech	Confidence
Hello	Interjection	0.98
Zhang	Proper noun	0.99+
Wei	Proper noun	0.99+
,	Punctuation	0.99+
I	Pronoun	0.99+
am	Verb	0.98
John	Proper noun	0.99+
.	Punctuation	0.99+
Your	Pronoun	0.99+
AnyCompany	Proper noun	0.99+

**► Application integration**

## Ausführen von Analyseaufträgen mit der Konsole

Sie können die Amazon Comprehend-Konsole verwenden, um asynchrone Analyseaufträge zu erstellen und zu verwalten. Ihr Auftrag analysiert in Amazon S3 gespeicherte Dokumente, um

Entitäten wie Ereignisse, Wortgruppen, Primärsprache, Stimmung oder persönlich identifizierbare Informationen (PII) zu finden.

So erstellen Sie einen Analyseauftrag

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon Comprehend-Konsole unter <https://console.aws.amazon.com/comprehend/>
2. Wählen Sie im linken Menü Analyseaufträge und dann Auftrag erstellen aus.
3. Geben Sie unter Auftragseinstellungen dem Analyseauftrag einen eindeutigen Namen.
4. Wählen Sie für Analysetyp einen der integrierten Analysetypen aus.

Wenn Sie Primäre Sprache oder Themenmodellierung wählen, können Sie den nächsten Schritt überspringen.

5. Je nach ausgewähltem Analysetyp zeigt die Konsole eines oder mehrere der folgenden zusätzlichen Felder an:

- Sprache ist für alle integrierten Analysetypen außer Primäre Sprache und Themenmodellierung erforderlich.

Wählen Sie die Sprache Ihrer Eingabedokumente aus.

- Zielereignistypen sind für den Ereignisanalysetyp erforderlich.


Wählen Sie die Arten von Ereignissen aus, die in Ihren Eingabedokumenten erkannt werden sollen. Weitere Informationen zu unterstützten Ereignistypen finden Sie unter [Ereignistypen](#).

- Die Einstellungen für die PII-Erkennung sind für den PII-Analysetyp erforderlich.

Wählen Sie den Ausgabemodus aus. Weitere Informationen zu den Einstellungen für die PII-Erkennung finden Sie unter [Erkennung von PII-Entitäten](#).

6. Geben Sie unter Eingabedaten an, wo sich die Eingabedokumente in Amazon S3 befinden:
  - Um Ihre eigenen Dokumente zu analysieren, wählen Sie Meine Dokumente und dann S3 durchsuchen, um den Pfad zu dem Bucket oder Ordner anzugeben, der Ihre Dateien enthält.
  - Um Beispiele zu analysieren, die von Amazon Comprehend bereitgestellt werden, wählen Sie Beispieldokumente aus. In diesem Fall verwendet Amazon Comprehend einen Bucket, der von verwaltet wird AWS, und Sie geben den Speicherort nicht an.
7. (Optional) Geben Sie für Eingabeformat eines der folgenden Formate für Ihre Eingabedateien an:

- Ein Dokument pro Datei – Jede Datei enthält ein Eingabedokument. Dies eignet sich am besten für Sammlungen großer Dokumente.
  - Ein Dokument pro Zeile – Die Eingabe besteht aus einer oder mehreren Dateien. Jede Zeile in einer Datei wird als Dokument betrachtet. Dies eignet sich am besten für kurze Dokumente, z. B. Social-Media-Beiträge. Jede Zeile muss mit einem Zeilenvorschub (LF, \n), einem Zeilenumlauf (CR, \r) oder beidem (CRLF, \r\n) enden. Sie können das UTF-8-Zeilentrennzeichen (u+2028) nicht verwenden, um eine Zeile zu beenden.
8. Wählen Sie unter Ausgabedaten die Option `S3 durchsuchen` aus. Wählen Sie den Amazon S3-Bucket oder -Ordner aus, in den Amazon Comprehend die Ausgabedaten schreiben soll, die von der Analyse erzeugt werden.
  9. (Optional) Um das Ausgabeergebnis Ihres Auftrags zu verschlüsseln, wählen Sie Verschlüsselung aus. Wählen Sie dann aus, ob Sie einen KMS-Schlüssel verwenden möchten, der dem aktuellen Konto zugeordnet ist, oder einen Schlüssel aus einem anderen Konto:
    - Wenn Sie einen Schlüssel verwenden, der dem aktuellen Konto zugeordnet ist, wählen Sie den Schlüsselalias oder die ID für die KMS-Schlüssel-ID aus.
    - Wenn Sie einen Schlüssel verwenden, der einem anderen Konto zugeordnet ist, geben Sie den ARN für den Schlüsselalias oder die ID unter KMS-Schlüssel-ID ein.

 Note

Weitere Informationen zum Erstellen und Verwenden von KMS-Schlüsseln und der zugehörigen Verschlüsselung finden Sie unter [Key Management Service \(KMS\)](#).

10. Geben Sie unter Zugriffsberechtigungen eine IAM-Rolle an, die:
  - Gewährt Lesezugriff auf den Amazon S3-Speicherort Ihrer Eingabedokumente.
  - Gewährt Schreibzugriff auf den Amazon S3-Speicherort Ihrer Ausgabedokumente.
  - Enthält eine Vertrauensrichtlinie, die es dem `comprehend.amazonaws.com` Service-Prinzipal ermöglicht, die Rolle zu übernehmen und ihre Berechtigungen zu erhalten.

Wenn Sie noch keine IAM-Rolle mit diesen Berechtigungen und einer geeigneten Vertrauensrichtlinie haben, wählen Sie Erstellen einer IAM-Rolle, um eine zu erstellen.

11. Wenn Sie das Formular ausgefüllt haben, wählen Sie Auftrag erstellen, um den Themenerkennungsauftrag zu erstellen und zu starten.

Der neue Auftrag wird in der Auftragsliste angezeigt, wobei das Statusfeld den Status des Auftrags anzeigt. Das Feld kann IN\_PROGRESS für einen Auftrag sein, der gerade verarbeitet wird, COMPLETED für einen Auftrag, der erfolgreich abgeschlossen wurde, und FAILED für einen Auftrag, der einen Fehler aufweist. Sie können auf einen Auftrag klicken, um weitere Informationen über den Auftrag zu erhalten, einschließlich Fehlermeldungen.

Wenn der Auftrag abgeschlossen ist, speichert Amazon Comprehend die Analyseergebnisse an dem Amazon S3-Ausgabespeicherort, den Sie für den Auftrag angegeben haben. Eine Beschreibung der Analyseergebnisse für jeden Insight-Typ finden Sie unter [Insights](#).

# Verwenden der Amazon Comprehend API

Die Amazon Comprehend API unterstützt Operationen zur Durchführung von (synchronen) Analysen in Echtzeit und Operationen zum Starten und Verwalten asynchroner Analysejobs.

Sie können die Amazon Comprehend API-Operatoren direkt verwenden, oder Sie können die CLI oder eines der SDKs verwenden. Die Beispiele in diesem Kapitel verwenden die CLI, das Python-SDK und das Java-SDK.

Um die AWS CLI und Python-Beispiele auszuführen, müssen Sie die installieren AWS CLI. Weitere Informationen finden Sie unter [Richten Sie das AWS Command Line Interface \(AWS CLI\) ein](#).

Um die Java-Beispiele auszuführen, müssen Sie den installieren AWS SDK for Java. Anweisungen zur Installation des SDK for Java finden Sie unter [AWS SDK for Java einrichten](#).

## Themen

- [Amazon Comprehend mit einem SDK verwenden AWS](#)
- [Echtzeitanalyse mit der API](#)
- [Asynchrone Analyseaufträge mit der API](#)

## Amazon Comprehend mit einem SDK verwenden AWS

AWS Software Development Kits (SDKs) sind für viele gängige Programmiersprachen verfügbar. Jedes SDK bietet eine API, Codebeispiele und Dokumentation, die es Entwicklern erleichtern, Anwendungen in ihrer bevorzugten Sprache zu erstellen.

SDK-Dokumentation	Codebeispiele
<a href="#">AWS SDK for C++</a>	<a href="#">AWS SDK for C++ Code-Beispiele</a>
<a href="#">AWS CLI</a>	<a href="#">AWS CLI Code-Beispiele</a>
<a href="#">AWS SDK for Go</a>	<a href="#">AWS SDK for Go Code-Beispiele</a>
<a href="#">AWS SDK for Java</a>	<a href="#">AWS SDK for Java Code-Beispiele</a>
<a href="#">AWS SDK for JavaScript</a>	<a href="#">AWS SDK for JavaScript Code-Beispiele</a>

SDK-Dokumentation	Codebeispiele
<a href="#">AWS SDK for Kotlin</a>	<a href="#">AWS SDK for Kotlin Code-Beispiele</a>
<a href="#">AWS SDK for .NET</a>	<a href="#">AWS SDK for .NET Code-Beispiele</a>
<a href="#">AWS SDK for PHP</a>	<a href="#">AWS SDK for PHP Code-Beispiele</a>
<a href="#">AWS Tools for PowerShell</a>	<a href="#">Tools für PowerShell Codebeispiele</a>
<a href="#">AWS SDK for Python (Boto3)</a>	<a href="#">AWS SDK for Python (Boto3) Code-Beispiele</a>
<a href="#">AWS SDK for Ruby</a>	<a href="#">AWS SDK for Ruby Code-Beispiele</a>
<a href="#">AWS SDK for Rust</a>	<a href="#">AWS SDK for Rust Code-Beispiele</a>
<a href="#">AWS SDK für SAP ABAP</a>	<a href="#">AWS SDK für SAP ABAP Code-Beispiele</a>
<a href="#">AWS SDK for Swift</a>	<a href="#">AWS SDK for Swift Code-Beispiele</a>

### Beispiel für die Verfügbarkeit

Sie können nicht finden, was Sie brauchen? Fordern Sie ein Codebeispiel an, indem Sie unten den Link [Provide feedback \(Feedback geben\)](#) auswählen.

## Echtzeitanalyse mit der API

Die folgenden Beispiele zeigen, wie Sie die Amazon Comprehend API für Echtzeitanalysen verwenden, indem Sie die AWS CLI und die AWS SDKs für .NET, Java und Python verwenden. Verwenden Sie die Beispiele, um mehr über die synchronen Operationen von Amazon Comprehend und als Bausteine für Ihre eigenen Anwendungen zu erfahren.

Die .NET-Beispiele in diesem Abschnitt verwenden das [AWS SDK for .NET](#). Sie können die [AWS Toolkit for Visual Studio](#) verwenden, um AWS Anwendungen mit .NET zu entwickeln. Es enthält hilfreiche Vorlagen und den AWS Explorer für die Bereitstellung von Anwendungen und die Verwaltung von -Services. Eine .NET-Entwicklerperspektive von AWS finden Sie im [AWS Handbuch für .NET-Entwickler](#).

## Themen

- [Erkennen der dominanten Sprache](#)
- [Erkennen benannter Entitäten](#)
- [Erkennen von Schlüsselphrasen](#)
- [Bestimmen der Stimmung](#)
- [Echtzeitanalyse für gezielte Stimmung](#)
- [Erkennen der Syntax](#)
- [Batch-APIs in Echtzeit](#)

## Erkennen der dominanten Sprache

Um die dominante Sprache zu ermitteln, die im Text verwendet wird, verwenden Sie die [-DetectDominantLanguage](#) Operation. Um die dominante Sprache in bis zu 25 Dokumenten in einem Batch zu erkennen, verwenden Sie die [-BatchDetectDominantLanguage](#) Operation. Weitere Informationen finden Sie unter [Batch-APIs in Echtzeit](#).

## Themen

- [Verwendung der AWS Command Line Interface](#)
- [Verwenden von AWS SDK for Java, SDK für Python oder AWS SDK for .NET](#)

## Verwendung der AWS Command Line Interface

Das folgende Beispiel zeigt die Verwendung der `-DetectDominantLanguage` Operation mit der AWS CLI.

Das Beispiel ist für Unix, Linux und macOS formatiert. Ersetzen Sie unter Windows den umgekehrten Schrägstrich (`\`), das Unix-Fortsetzungszeichen, am Ende jeder Zeile durch ein Caret-Zeichen oder Zirkumflex (`^`).

```
aws comprehend detect-dominant-language \  
  --region region \  
  --text "It is raining today in Seattle."
```

Amazon Comprehend antwortet wie folgt:

```
{
```



```
"Languages": [  
  {  
    "LanguageCode": "en",  
    "Score": 0.9793661236763  
  }  
]
```

Verwenden von AWS SDK for Java, SDK für Python oder AWS SDK for .NET

SDK-Beispiele zur Bestimmung der dominanten Sprache finden Sie unter [Verwendung DetectDominantLanguage mit einem AWS SDK oder CLI](#).

## Erkennen benannter Entitäten

Um die benannten Entitäten in einem Dokument zu ermitteln, verwenden Sie die [-DetectEntities](#) Operation. Um Entitäten in bis zu 25 Dokumenten in einem Batch zu erkennen, verwenden Sie die [-BatchDetectEntities](#) Operation. Weitere Informationen finden Sie unter [Batch-APIs in Echtzeit](#).

Themen

- [Verwendung der AWS Command Line Interface](#)
- [Verwenden von AWS SDK for Java, SDK für Python oder AWS SDK for .NET](#)

## Verwendung der AWS Command Line Interface

Das folgende Beispiel zeigt die Verwendung der `-DetectEntities` Operation unter Verwendung der AWS CLI. Sie müssen die Sprache des Eingabetexts angeben.

Das Beispiel ist für Unix, Linux und macOS formatiert. Ersetzen Sie unter Windows den umgekehrten Schrägstrich (`\`), das Unix-Fortsetzungszeichen, am Ende jeder Zeile durch ein Caret-Zeichen oder Zirkumflex (`^`).

```
aws comprehend detect-entities \  
  --region region \  
  --language-code "en" \  
  --text "It is raining today in Seattle."
```

Amazon Comprehend antwortet wie folgt:

```
{
  "Entities": [
    {
      "Text": "today",
      "Score": 0.97,
      "Type": "DATE",
      "BeginOffset": 14,
      "EndOffset": 19
    },
    {
      "Text": "Seattle",
      "Score": 0.95,
      "Type": "LOCATION",
      "BeginOffset": 23,
      "EndOffset": 30
    }
  ],
  "LanguageCode": "en"
}
```

## Verwenden von AWS SDK for Java, SDK für Python oder AWS SDK for .NET

SDK-Beispiele zur Bestimmung der dominanten Sprache finden Sie unter [Verwendung DetectEntities mit einem AWS SDK oder CLI](#).

## Erkennen von Schlüsselphrasen

Um die im Text verwendeten Schlüsselnamenphrasen zu ermitteln, verwenden Sie die [-DetectKeyPhrases](#)Operation. Um die Schlüsselnamenphrasen in bis zu 25 Dokumenten in einem Batch zu erkennen, verwenden Sie die [-BatchDetectKeyPhrases](#)Operation. Weitere Informationen finden Sie unter [Batch-APIs in Echtzeit](#).

### Themen

- [Verwendung der AWS Command Line Interface](#)
- [Verwenden von AWS SDK for Java, SDK für Python oder AWS SDK for .NET](#)

## Verwendung der AWS Command Line Interface

Das folgende Beispiel zeigt die Verwendung der `-DetectKeyPhrases`Operation mit der AWS CLI. Sie müssen die Sprache des Eingabetexts angeben.

Das Beispiel ist für Unix, Linux und macOS formatiert. Ersetzen Sie unter Windows den umgekehrten Schrägstrich (\), das Unix-Fortsetzungszeichen, am Ende jeder Zeile durch ein Caret-Zeichen oder Zirkumflex (^).

```
aws comprehend detect-key-phrases \  
  --region region \  
  --language-code "en" \  
  --text "It is raining today in Seattle."
```

Amazon Comprehend antwortet wie folgt:

```
{  
  "LanguageCode": "en",  
  "KeyPhrases": [  
    {  
      "Text": "today",  
      "Score": 0.89,  
      "BeginOffset": 14,  
      "EndOffset": 19  
    },  
    {  
      "Text": "Seattle",  
      "Score": 0.91,  
      "BeginOffset": 23,  
      "EndOffset": 30  
    }  
  ]  
}
```

Verwenden von AWS SDK for Java, SDK für Python oder AWS SDK for .NET

SDK-Beispiele, die Schlüsselphrasen erkennen, finden Sie unter [Verwendung DetectKeyPhrases mit einem AWS SDK oder CLI](#).

## Bestimmen der Stimmung

Amazon Comprehend bietet die folgenden API-Operationen zum Analysieren der Stimmung:

- [DetectSentiment](#) – Bestimmt die gesamte neuronale Stimmung eines Dokuments.
- [BatchDetectSentiment](#) – Ermitteln Sie die Gesamtstimmung in bis zu 25 Dokumenten in einem Batch. Weitere Informationen finden Sie unter [Batch-APIs in Echtzeit](#).

- [StartSentimentDetectionJob](#) – Startet einen asynchronen Stimmungserkennungsauftrag für eine Sammlung von Dokumenten.
- [ListSentimentDetectionJobs](#) – Gibt die Liste der von Ihnen übermittelten Aufträge zur Stimmungserkennung zurück.
- [DescribeSentimentDetectionJob](#) – Ruft die Eigenschaften (einschließlich Status) ab, die dem angegebenen Stimmungserkennungsauftrag zugeordnet sind.
- [StopSentimentDetectionJob](#) – Bricht den angegebenen Stimmungsauftrag in Bearbeitung ab.

## Themen

- [Verwenden des AWS Command Line Interface](#)
- [Verwenden von AWS SDK for Java, SDK für Python oder AWS SDK for .NET](#)

## Verwenden des AWS Command Line Interface

Das folgende Beispiel zeigt die Verwendung der `-DetectSentimentOperation` mit der AWS CLI. In diesem Beispiel wird die Sprache des Eingabetexts angegeben.

Das Beispiel ist für Unix, Linux und macOS formatiert. Ersetzen Sie unter Windows den umgekehrten Schrägstrich (`\`), das Unix-Fortsetzungszeichen, am Ende jeder Zeile durch ein Caret-Zeichen oder Zirkumflex (`^`).

```
aws comprehend detect-sentiment \  
  --region region \  
  --language-code "en" \  
  --text "It is raining today in Seattle."
```

Amazon Comprehend antwortet wie folgt:

```
{  
  "SentimentScore": {  
    "Mixed": 0.014585512690246105,  
    "Positive": 0.31592071056365967,  
    "Neutral": 0.5985543131828308,  
    "Negative": 0.07093945890665054  
  },  
  "Sentiment": "NEUTRAL",  
  "LanguageCode": "en"  
}
```

## Verwenden von AWS SDK for Java, SDK für Python oder AWS SDK for .NET

SDK-Beispiele, die die Stimmung des Eingabetexts bestimmen, finden Sie unter [Verwendung DetectSentiment mit einem AWS SDK oder CLI](#).

## Echtzeitanalyse für gezielte Stimmung

Amazon Comprehend bietet die folgenden API-Operationen für gezielte Stimmungs-Echtzeitanalysen:

- [DetectTargetedSentiment](#) – Analysiert die Stimmung der in einem Dokument erwähnten Entitäten.
- [BatchDetectTargetedSentiment](#) – Analysiert die gezielte Stimmung für bis zu 25 Dokumente in einem Batch. Weitere Informationen finden Sie unter [Batch-APIs in Echtzeit](#).

Wenn der analysierte Text keine gezielte Stimmung enthält [Entitätstypen](#), gibt die API ein leeres Entitäts-Array zurück.

## Verwenden des AWS Command Line Interface

Das folgende Beispiel zeigt die Verwendung der `-DetectTargetedSentimentOperation` mit der AWS CLI. In diesem Beispiel wird die Sprache des Eingabetexts angegeben.

Das Beispiel ist für Unix, Linux und macOS formatiert. Ersetzen Sie unter Windows den umgekehrten Schrägstrich (`\`), das Unix-Fortsetzungszeichen, am Ende jeder Zeile durch ein Caret-Zeichen oder Zirkumflex (`^`).

```
aws comprehend detect-targeted-sentiment \  
  --region region \  
  --language-code "en" \  
  --text "The burger was cooked perfectly but it was cold. The service was OK."
```

Amazon Comprehend antwortet wie folgt:

```
{  
  "Entities": [  
    {  
      "DescriptiveMentionIndex": [  
        0  
      ],  
      "Mentions": [  
        {
```

```
    "BeginOffset": 4,
    "EndOffset": 10,
    "Score": 1,
    "GroupScore": 1,
    "Text": "burger",
    "Type": "OTHER",
    "MentionSentiment": {
      "Sentiment": "POSITIVE",
      "SentimentScore": {
        "Mixed": 0.001515,
        "Negative": 0.000822,
        "Neutral": 0.000243,
        "Positive": 0.99742
      }
    }
  },
  {
    "BeginOffset": 36,
    "EndOffset": 38,
    "Score": 0.999843,
    "GroupScore": 0.999661,
    "Text": "it",
    "Type": "OTHER",
    "MentionSentiment": {
      "Sentiment": "NEGATIVE",
      "SentimentScore": {
        "Mixed": 0,
        "Negative": 0.999996,
        "Neutral": 0.000004,
        "Positive": 0
      }
    }
  }
]
},
{
  "DescriptiveMentionIndex": [
    0
  ],
  "Mentions": [
    {
      "BeginOffset": 53,
      "EndOffset": 60,
      "Score": 1,
```

```
    "GroupScore": 1,
    "Text": "service",
    "Type": "ATTRIBUTE",
    "MentionSentiment": {
      "Sentiment": "NEUTRAL",
      "SentimentScore": {
        "Mixed": 0.000033,
        "Negative": 0.000089,
        "Neutral": 0.993325,
        "Positive": 0.006553
      }
    }
  }
]
}
```

## Erkennen der Syntax

Um Text zu analysieren, um die einzelnen Wörter zu extrahieren und die Sprachteile für jedes Wort zu bestimmen, verwenden Sie die [-DetectSyntax](#) Operation. Um die Syntax von bis zu 25 Dokumenten in einem Batch zu analysieren, verwenden Sie die [-BatchDetectSyntax](#) Operation. Weitere Informationen finden Sie unter [Batch-APIs in Echtzeit](#).

### Themen

- [Verwenden der AWS Command Line Interface](#).
- [Verwenden von AWS SDK for Java, SDK für Python oder AWS SDK for .NET](#)

## Verwenden der AWS Command Line Interface.

Das folgende Beispiel zeigt die Verwendung der `-DetectSyntax` Operation mit der AWS CLI. In diesem Beispiel wird die Sprache des Eingabetexts angegeben.

Das Beispiel ist für Unix, Linux und macOS formatiert. Ersetzen Sie unter Windows den umgekehrten Schrägstrich (`\`), das Unix-Fortsetzungszeichen, am Ende jeder Zeile durch ein Caret-Zeichen oder Zirkumflex (`^`).

```
aws comprehend detect-syntax \  
  --region region \  
  --text "service"
```

```
--language-code "en" \  
--text "It is raining today in Seattle."
```

Amazon Comprehend antwortet wie folgt:

```
{  
  "SyntaxTokens": [  
    {  
      "Text": "It",  
      "EndOffset": 2,  
      "BeginOffset": 0,  
      "PartOfSpeech": {  
        "Tag": "PRON",  
        "Score": 0.8389829397201538  
      },  
      "TokenId": 1  
    },  
    {  
      "Text": "is",  
      "EndOffset": 5,  
      "BeginOffset": 3,  
      "PartOfSpeech": {  
        "Tag": "AUX",  
        "Score": 0.9189288020133972  
      },  
      "TokenId": 2  
    },  
    {  
      "Text": "raining",  
      "EndOffset": 13,  
      "BeginOffset": 6,  
      "PartOfSpeech": {  
        "Tag": "VERB",  
        "Score": 0.9977611303329468  
      },  
      "TokenId": 3  
    },  
    {  
      "Text": "today",  
      "EndOffset": 19,  
      "BeginOffset": 14,  
      "PartOfSpeech": {  
        "Tag": "NOUN",
```



```
        "Score": 0.9993606209754944
    },
    "TokenId": 4
},
{
    "Text": "in",
    "EndOffset": 22,
    "BeginOffset": 20,
    "PartOfSpeech": {
        "Tag": "ADP",
        "Score": 0.9999061822891235
    },
    "TokenId": 5
},
{
    "Text": "Seattle",
    "EndOffset": 30,
    "BeginOffset": 23,
    "PartOfSpeech": {
        "Tag": "PROPN",
        "Score": 0.9940338730812073
    },
    "TokenId": 6
},
{
    "Text": ".",
    "EndOffset": 31,
    "BeginOffset": 30,
    "PartOfSpeech": {
        "Tag": "PUNCT",
        "Score": 0.9999997615814209
    },
    "TokenId": 7
}
]
}
```

## Verwenden von AWS SDK for Java, SDK für Python oder AWS SDK for .NET

SDK-Beispiele, die die Syntax von Eingabetext erkennen, finden Sie unter [Verwendung DetectSyntax mit einem AWS SDK oder CLI](#).

## Batch-APIs in Echtzeit

Um Batches mit bis zu 25 Dokumenten zu senden, können Sie die Amazon Comprehend-Echtzeit-Batchoperationen verwenden. Das Aufrufen eines Batchvorgangs ist identisch mit dem Aufrufen der einzelnen Dokument-APIs für jedes Dokument in der Anforderung. Die Verwendung der Batch-APIs kann zu einer besseren Leistung für Ihre Anwendungen führen. Weitere Informationen finden Sie unter [Synchrone Verarbeitung mehrerer Dokumente](#).

### Themen

- [Batchverarbeitung mit der AWS CLI](#)
- [Batchverarbeitung mit der AWS SDK for .NET](#)

## Batchverarbeitung mit der AWS CLI

Diese Beispiele zeigen, wie Sie die Batch-API-Operationen mithilfe der verwenden AWS Command Line Interface. Alle Operationen außer `BatchDetectDominantLanguage` verwenden die folgende JSON-Datei namens `process.json` als Eingabe. Für diesen Vorgang ist die `LanguageCode` Entität nicht enthalten.

Das dritte Dokument in der JSON-Datei ("\$\$\$\$\$\$\$\$") verursacht während der Batchverarbeitung einen Fehler. Sie ist enthalten, sodass die Operationen eine [BatchItemError](#) in der Antwort enthalten.

```
{
  "LanguageCode": "en",
  "TextList": [
    "I have been living in Seattle for almost 4 years",
    "It is raining today in Seattle",
    "$$$$$$$$"
  ]
}
```

Die Beispiele sind für Unix, Linux und macOS formatiert. Ersetzen Sie unter Windows den umgekehrten Schrägstrich (\), das Unix-Fortsetzungszeichen, am Ende jeder Zeile durch ein Caret-Zeichen oder Zirkumflex (^).

### Themen

- [Erkennen der dominanten Sprache mithilfe eines Batches \(AWS CLI\)](#)
- [Entitäten mithilfe eines Batches erkennen \(AWS CLI\)](#)

- [Erkennen von Schlüsselphrasen mithilfe eines Batches \(AWS CLI\)](#)
- [Erkennen der Stimmung mithilfe eines Batches \(AWS CLI\)](#)

## Erkennen der dominanten Sprache mithilfe eines Batches (AWS CLI)

Die `-BatchDetectDominantLanguage` Operation bestimmt die dominante Sprache jedes Dokuments in einem Batch. Eine Liste der Sprachen, die Amazon Comprehend erkennen kann, finden Sie unter [Dominante Sprache](#). Der folgende AWS CLI Befehl ruft die `-BatchDetectDominantLanguage` Operation auf.

```
aws comprehend batch-detect-dominant-language \  
  --endpoint endpoint \  
  --region region \  
  --cli-input-json file://path to input file/process.json
```

Im Folgenden finden Sie die Antwort des `-BatchDetectDominantLanguage` Vorgangs:

```
{  
  "ResultList": [  
    {  
      "Index": 0,  
      "Languages": [  
        {  
          "LanguageCode": "en",  
          "Score": 0.99  
        }  
      ]  
    },  
    {  
      "Index": 1,  
      "Languages": [  
        {  
          "LanguageCode": "en",  
          "Score": 0.82  
        }  
      ]  
    }  
  ],  
  "ErrorList": [  
    {  
      "Index": 2,
```

```
    "ErrorCode": "InternalServerError",
    "ErrorMessage": "Unexpected Server Error. Please try again."
  }
]
```

### Entitäten mithilfe eines Batches erkennen (AWS CLI)

Verwenden Sie die [BatchDetectEntities](#) Operation, um die Entitäten zu finden, die in einem Stapel von Dokumenten vorhanden sind. Weitere Informationen über Entitys finden Sie unter [Entitäten](#). Der folgende AWS CLI Befehl ruft die `-BatchDetectEntities` Operation auf.

```
aws comprehend batch-detect-entities \  
  --endpoint endpoint \  
  --region region \  
  --cli-input-json file://path to input file/process.json
```

### Erkennen von Schlüsselphrasen mithilfe eines Batches (AWS CLI)

Die [BatchDetectKeyPhrases](#) Operation gibt die Schlüsselnamenphrasen in einem Stapel von Dokumenten zurück. Der folgende AWS CLI Befehl ruft die `-BatchDetectKeyNounPhrases` Operation auf.

```
aws comprehend batch-detect-key-phrases \  
  --endpoint endpoint \  
  --region region \  
  --cli-input-json file://path to input file/process.json
```

### Erkennen der Stimmung mithilfe eines Batches (AWS CLI)

Erkennen Sie die Gesamtstimmung eines Dokumentenstapels mithilfe der [BatchDetectSentiment](#) Operation. Der folgende AWS CLI Befehl ruft die `-BatchDetectSentiment` Operation auf.

```
aws comprehend batch-detect-sentiment \  
  --endpoint endpoint \  
  --region region \  
  --cli-input-json file://path to input file/process.json
```

## Batchverarbeitung mit der AWS SDK for .NET

Das folgende Beispielprogramm zeigt, wie Sie die [BatchDetectEntities](#) Operation mit dem `Amazon.Comprehend` SDK for .NET verwenden. Die Antwort vom Server enthält ein [BatchDetectEntitiesItemResult](#) Objekt für jedes Dokument, das erfolgreich verarbeitet wurde. Wenn bei der Verarbeitung eines Dokuments ein Fehler auftritt, befindet sich in der Fehlerliste in der Antwort ein Datensatz. Das Beispiel ruft jedes der Dokumente mit einem Fehler ab und sendet sie erneut.

Das .NET-Beispiel in diesem Abschnitt verwendet die [AWS SDK for .NET](#). Sie können die [AWS Toolkit for Visual Studio](#) verwenden, um AWS Anwendungen mit .NET zu entwickeln. Es enthält hilfreiche Vorlagen und den AWS Explorer für die Bereitstellung von Anwendungen und die Verwaltung von -Services. Eine .NET-Entwicklerperspektive von AWS finden Sie im [AWS Handbuch für .NET-Entwickler](#).

```
using System;
using System.Collections.Generic;
using Amazon.Comprehend;
using Amazon.Comprehend.Model;

namespace Comprehend
{
    class Program
    {
        // Helper method for printing properties
        static private void PrintEntity(Entity entity)
        {
            Console.WriteLine("    Text: {0}, Type: {1}, Score: {2}, BeginOffset: {3}
EndOffset: {4}",
                entity.Text, entity.Type, entity.Score, entity.BeginOffset,
entity.EndOffset);
        }

        static void Main(string[] args)
        {
            AmazonComprehendClient comprehendClient = new
AmazonComprehendClient(Amazon.RegionEndpoint.USWest2);

            List<String> textList = new List<String>()
            {
                { "I love Seattle" },
                { "Today is Sunday" },
                { "Tomorrow is Monday" },
            }
        }
    }
}
```

```
        { "I love Seattle" }
    };

    // Call detectEntities API
    Console.WriteLine("Calling BatchDetectEntities");
    BatchDetectEntitiesRequest batchDetectEntitiesRequest = new
BatchDetectEntitiesRequest()
    {
        TextList = textList,
        LanguageCode = "en"
    };
    BatchDetectEntitiesResponse batchDetectEntitiesResponse =
comprehendClient.BatchDetectEntities(batchDetectEntitiesRequest);

    foreach (BatchDetectEntitiesItemResult item in
batchDetectEntitiesResponse.ResultList)
    {
        Console.WriteLine("Entities in {0}:", textList[item.Index]);
        foreach (Entity entity in item.Entities)
            PrintEntity(entity);
    }

    // check if we need to retry failed requests
    if (batchDetectEntitiesResponse.ErrorList.Count != 0)
    {
        Console.WriteLine("Retrying Failed Requests");
        List<String> textToRetry = new List<String>();
        foreach (BatchItemError errorItem in
batchDetectEntitiesResponse.ErrorList)
            textToRetry.Add(textList[errorItem.Index]);

        batchDetectEntitiesRequest = new BatchDetectEntitiesRequest()
        {
            TextList = textToRetry,
            LanguageCode = "en"
        };

        batchDetectEntitiesResponse =
comprehendClient.BatchDetectEntities(batchDetectEntitiesRequest);

        foreach (BatchDetectEntitiesItemResult item in
batchDetectEntitiesResponse.ResultList)
        {
            Console.WriteLine("Entities in {0}:", textList[item.Index]);
```

```
        foreach (Entity entity in item.Entities)
            PrintEntity(entity);
    }
}
Console.WriteLine("End of DetectEntities");
}
}
```

## Asynchrone Analyseaufträge mit der API

In den folgenden Beispielen werden die asynchronen APIs von Amazon Comprehend verwendet, um Analyseaufträge mithilfe der zu erstellen und zu verwalten AWS CLI.

### Themen

- [Asynchrone Analyse für Amazon Comprehend Insights](#)
- [Asynchrone Analyse für gezielte Stimmung](#)
- [Asynchrone Analyse zur Ereigniserkennung](#)
- [Asynchrone Analyse für die Themenmodellierung](#)

## Asynchrone Analyse für Amazon Comprehend Insights

In den folgenden Abschnitten wird die Amazon Comprehend API verwendet, um asynchrone Operationen zur Analyse von Amazon Comprehend Insights auszuführen.

### Themen

- [Voraussetzungen](#)
- [Starten eines Analyseauftrags](#)
- [Überwachen von Analyseaufträgen](#)
- [Abrufen von Analyseergebnissen](#)

## Voraussetzungen

Dokumente müssen in Textdateien im UTF-8-formatte. Sie können Ihre Dokumente in zwei Formaten einreichen. Das Format, das Sie verwenden, hängt von der Art der Dokumente ab, die Sie analysieren möchten, wie in der folgenden Tabelle beschrieben.

Beschreibung	Format
Jede Datei enthält ein Eingabedokument. Dies eignet sich am besten für Sammlungen großer Dokumente.	Ein Dokument pro Datei
Die Eingabe besteht aus einer oder mehreren Dateien. Jede Zeile in einer Datei wird als Dokument betrachtet. Dies eignet sich am besten für kurze Dokumente, z. B. Social-Media-Beiträge.  Jede Zeile muss mit einem Zeilenvorschub (LF, \n), einem Zeilenumlauf (CR, \r) oder beidem (CRLF, \r\n) enden. Sie können das UTF-8-Zeilentrennzeichen (u+2028) nicht verwenden, um eine Zeile zu beenden.	Ein Dokument pro Zeile

Wenn Sie einen Analyseauftrag starten, geben Sie den S3-Speicherort für Ihre Eingabedaten an. Der URI muss sich in derselben AWS Region wie der API-Endpunkt befinden, den Sie aufrufen. Der URI kann auf eine einzelne Datei verweisen oder das Präfix für eine Sammlung von Datendateien sein. Weitere Informationen finden Sie unter [-InputDataConfig](#)Datentyp.

Sie müssen Amazon Comprehend Zugriff auf den Amazon S3-Bucket gewähren, der Ihre Dokumentenerfassungs- und Ausgabedateien enthält. Weitere Informationen finden Sie unter [Rollenbasierte Berechtigungen, die für asynchrone Operationen erforderlich sind](#).

## Starten eines Analyseauftrags

Um einen Analyseauftrag einzureichen, verwenden Sie entweder die Amazon Comprehend-Konsole oder die entsprechende `Start*` Operation:

- [StartDominantLanguageDetectionJob](#) – Starten Sie einen Auftrag, um die vorherrschende Sprache in jedem Dokument in der Sammlung zu erkennen. Weitere Informationen zur dominierenden Sprache in einem Dokument finden Sie unter [Dominante Sprache](#).
- [StartEntitiesDetectionJob](#) – Starten Sie einen Auftrag, um Entitäten in jedem Dokument in der Sammlung zu erkennen. Weitere Informationen über Entitäts finden Sie unter [Entitäten](#).



- [StartKeyPhrasesDetectionJob](#) – Starten Sie einen Auftrag, um Schlüsselphrasen in jedem Dokument in der Sammlung zu erkennen. Weitere Informationen zu Schlüsselphrasen finden Sie unter [Schlüsselphrasen](#).
- [StartPiiEntitiesDetectionJob](#) – Starten Sie einen Auftrag, um persönlich identifizierbare Informationen (PII) in jedem Dokument in der Sammlung zu erkennen. Weitere Informationen zu PII finden Sie unter [Erkennung von PII-Entitäten](#).
- [StartSentimentDetectionJob](#) – Starten Sie einen Auftrag, um die Stimmung in jedem Dokument in der Sammlung zu erkennen. Weitere Informationen zu Stimmungen finden Sie unter [Stimmung](#).

## Überwachen von Analyseaufträgen

Der `Start*` Vorgang gibt eine ID zurück, mit der Sie den Fortschritt des Auftrags überwachen können.

Um den Fortschritt mithilfe der API zu überwachen, verwenden Sie eine von zwei Operationen, je nachdem, ob Sie den Fortschritt eines einzelnen oder mehrerer Aufträge überwachen möchten.

Verwenden Sie die `-Describe*` Operationen, um den Fortschritt eines einzelnen Analyseauftrags zu überwachen. Sie geben die Auftrags-ID an, die von der `Start*` Operation zurückgegeben wird. Die Antwort des `-Describe*` Vorgangs enthält das `JobStatus` Feld mit dem Status des Auftrags.

Um den Fortschritt mehrerer Analyseaufträge zu überwachen, verwenden Sie die `-List*` Operationen. `-List*` Operationen geben eine Liste von Aufträgen zurück, die Sie an Amazon Comprehend übermittelt haben. Die Antwort enthält ein `JobStatus` Feld für jeden Auftrag, das Ihnen den Status des Auftrags mitteilt.

Wenn das Statusfeld auf `COMPLETED` oder gesetzt ist `FAILED`, wurde die Auftragsverarbeitung abgeschlossen.

Um den Status einzelner Aufträge abzurufen, verwenden Sie den `Describe*` Vorgang für die Analyse, die Sie durchführen.

- [DescribeDominantLanguageDetectionJob](#)
- [DescribeEntitiesDetectionJob](#)
- [DescribeKeyPhrasesDetectionJob](#)
- [DescribePiiEntitiesDetectionJob](#)
- [DescribeSentimentDetectionJob](#)

Um den Status mehrerer Aufträge abzurufen, verwenden Sie die `-List*` Operation für die Analyse, die Sie durchführen.

- [ListDominantLanguageDetectionJobs](#)
- [ListEntitiesDetectionJobs](#)
- [ListKeyPhrasesDetectionJobs](#)
- [ListPiiEntitiesDetectionJobs](#)
- [ListSentimentDetectionJobs](#)

Um die Ergebnisse auf Aufträge zu beschränken, die bestimmten Kriterien entsprechen, verwenden Sie den `Filter` Parameter der `-List*` Operationen. Sie können nach dem Auftragsnamen, dem Auftragsstatus sowie dem Datum und der Uhrzeit filtern, zu der der Auftrag übermittelt wurde. Weitere Informationen finden Sie im `Filter` Parameter für jede der `List*` Operationen in der Amazon Comprehend API-Referenz.

## Abrufen von Analyseergebnissen

Nachdem ein Analyseauftrag abgeschlossen ist, verwenden Sie eine `-Describe*` Operation, um den Speicherort der Ergebnisse abzurufen. Wenn der Auftragsstatus lautet `COMPLETED`, enthält die Antwort ein Feld `OutputDataConfig`, das ein Feld mit dem Amazon S3-Speicherort der Ausgabedatei enthält. Die Datei `output.tar.gz` ist ein komprimiertes Archiv, das die Ergebnisse der Analyse enthält.

Wenn der Status eines Auftrags lautet `FAILED`, enthält die Antwort ein `Message` Feld, das den Grund beschreibt, warum der Analyseauftrag nicht erfolgreich abgeschlossen wurde.

Um den Status einzelner Aufträge abzurufen, verwenden Sie die entsprechende `Describe*` Operation:

- [DescribeDominantLanguageDetectionJob](#)
- [DescribeEntitiesDetectionJob](#)
- [DescribeKeyPhrasesDetectionJob](#)
- [DescribeSentimentDetectionJob](#)

Die Ergebnisse werden in einer einzigen Datei mit einer JSON-Struktur für jedes Dokument zurückgegeben. Jede Antwortdatei enthält auch Fehlermeldungen für jeden Auftrag, bei dem das Statusfeld auf `FAILED` gesetzt ist.

Jeder der folgenden Abschnitte zeigt Beispiele für die Ausgabe für die beiden Eingabeformate.

### Abrufen dominanter Spracherkennungsergebnisse

Im Folgenden finden Sie ein Beispiel für eine Ausgabedatei aus einer Analyse, die die dominante Sprache erkannt hat. Das Format der Eingabe ist ein Dokument pro Zeile. Weitere Informationen finden Sie unter dem Vorgang [DetectDominantLanguage](#).

```
{
  "File": "0_doc",
  "Languages": [
    {
      "LanguageCode": "en",
      "Score": 0.9514502286911011
    },
    {
      "LanguageCode": "de",
      "Score": 0.02374090999364853
    },
    {
      "LanguageCode": "nl",
      "Score": 0.003208699868991971
    }
  ],
  "Line": 0
},
{
  "File": "1_doc",
  "Languages": [
    {
      "LanguageCode": "en",
      "Score": 0.9822712540626526
    },
    {
      "LanguageCode": "de",
      "Score": 0.002621392020955682
    },
    {
      "LanguageCode": "es",
      "Score": 0.002386554144322872
    }
  ],
  "Line": 1
}
```

Im Folgenden finden Sie ein Beispiel für die Ausgabe einer Analyse, bei der das Format der Eingabe ein Dokument pro Datei ist:

```
{
  "File": "small_doc",
  "Languages": [
    {
      "LanguageCode": "en",
      "Score": 0.9728053212165833
    },
    {
      "LanguageCode": "de",
      "Score": 0.007670710328966379
    },
    {
      "LanguageCode": "es",
      "Score": 0.0028472368139773607
    }
  ]
},
{
  "File": "huge_doc",
  "Languages": [
    {
      "LanguageCode": "en",
      "Score": 0.984955906867981
    },
    {
      "LanguageCode": "de",
      "Score": 0.0026436643674969673
    },
    {
      "LanguageCode": "fr",
      "Score": 0.0014206881169229746
    }
  ]
}
```

### Abrufen der Ergebnisse der Entitätserkennung

Im Folgenden finden Sie ein Beispiel für eine Ausgabedatei aus einer Analyse, die Entitäten in Dokumenten erkannt hat. Das Format der Eingabe ist ein Dokument pro Zeile. Weitere Informationen finden Sie unter dem Vorgang [DetectEntities](#). Die Ausgabe enthält zwei Fehlermeldungen, eine für ein zu langes Dokument und eine für ein Dokument, das nicht im UTF-8-Format vorliegt.

```
{
  "File": "50_docs",
  "Line": 0,
  "Entities": [
    {
      "BeginOffset": 0,
      "EndOffset": 22,
      "Score": 0.9763959646224976,
      "Text": "Cluj-NapocaCluj-Napoca",
      "Type": "LOCATION"
    }
  ]
},
{
  "File": "50_docs",
  "Line": 1,
  "Entities": [
    {
      "BeginOffset": 11,
      "EndOffset": 15,
      "Score": 0.9615424871444702,
      "Text": "Maat",
      "Type": "PERSON"
    }
  ]
},
{
  "File": "50_docs",
  "Line": 2,
  "ErrorCode": "DOCUMENT_SIZE_EXCEEDED",
  "ErrorMessage": "Document size exceeds maximum size limit 102400 bytes."
},
{
  "File": "50_docs",
  "Line": 3,
  "ErrorCode": "UNSUPPORTED_ENCODING",
  "ErrorMessage": "Document is not in UTF-8 format and all subsequent lines are ignored."
}
```

Im Folgenden finden Sie ein Beispiel für die Ausgabe einer Analyse, bei der das Format der Eingabe ein Dokument pro Datei ist. Die Ausgabe enthält zwei Fehlermeldungen, eine für ein zu langes Dokument und eine für ein Dokument, das nicht im UTF-8-Format vorliegt.

```
{
  "File": "non_utf8.txt",
  "ErrorCode": "UNSUPPORTED_ENCODING",
  "ErrorMessage": "Document is not in UTF-8 format and all subsequent line are ignored."
}
{
  "File": "small_doc",
  "Entities": [
    {
      "BeginOffset": 0,
      "EndOffset": 4,
      "Score": 0.645766019821167,
      "Text": "Maat",
      "Type": "PERSON"
    }
  ]
}
{
  "File": "huge_doc",
  "ErrorCode": "DOCUMENT_SIZE_EXCEEDED",
  "ErrorMessage": "Document size exceeds size limit 102400 bytes."
}
```

### Abrufen der Erkennungsergebnisse für Schlüsselphrasen

Im Folgenden finden Sie ein Beispiel für eine Ausgabedatei aus einer Analyse, die Schlüsselphrasen in einem Dokument erkannt hat. Das Format der Eingabe ist ein Dokument pro Zeile. Weitere Informationen finden Sie unter dem Vorgang [DetectKeyPhrases](#).

```
{
  "File": "50_docs",
  "KeyPhrases": [
    {
      "BeginOffset": 0,
      "EndOffset": 22,
      "Score": 0.8948641419410706,
      "Text": "Cluj-NapocaCluj-Napoca"
    },
    {
      "BeginOffset": 45,
      "EndOffset": 49,
      "Score": 0.9989854693412781,
      "Text": "Cluj"
    }
  ],
  "Line": 0
}
```

Im Folgenden finden Sie ein Beispiel für die Ausgabe einer Analyse, bei der das Format der Eingabe ein Dokument pro Datei ist.

```
{
  "File": "1_doc",
  "KeyPhrases": [
    {
      "BeginOffset": 0,
      "EndOffset": 22,
      "Score": 0.8948641419410706,
      "Text": "Cluj-NapocaCluj-Napoca"
    },
    {
      "BeginOffset": 45,
      "EndOffset": 49,
      "Score": 0.9989854693412781,
      "Text": "Cluj"
    }
  ]
}
```

### Abrufen der Erkennungsergebnisse für persönlich identifizierbare Informationen (PII)

Im Folgenden finden Sie ein Beispiel für eine Ausgabedatei aus einem Analyseauftrag, der PII-Entitäten in Dokumenten erkannt hat. Das Format der Eingabe ist ein Dokument pro Zeile.

```
{
  "Entities": [
    {
      "Type": "NAME",
      "BeginOffset": 40,
      "EndOffset": 69,
      "Score": 0.999995
    },
    {
      "Type": "ADDRESS",
      "BeginOffset": 247,
      "EndOffset": 253,
      "Score": 0.998828
    },
    {
      "Type": "BANK_ACCOUNT_NUMBER",
      "BeginOffset": 406,
      "EndOffset": 411,
      "Score": 0.693283
    }
  ],
  "File": "doc.txt",
  "Entities": [
    {
      "Type": "SSN",
      "BeginOffset": 1114,
      "EndOffset": 1124,
      "Score": 0.999999
    },
    {
      "Type": "EMAIL",
      "BeginOffset": 3742,
      "EndOffset": 3775,
      "Score": 0.999993
    },
    {
      "Type": "PIN",
      "BeginOffset": 4098,
      "EndOffset": 4102,
      "Score": 0.999995
    }
  ],
  "File": "doc.txt",
  "Line": 1
}
```

Im Folgenden finden Sie ein Beispiel für die Ausgabe einer Analyse, bei der das Format der Eingabe ein Dokument pro Datei ist.

```
{"Entities": [{"Type": "NAME", "BeginOffset": 40, "EndOffset": 69, "Score": 0.999995}, {"Type": "ADDRESS", "BeginOffset": 247, "EndOffset": 253, "Score": 0.998828}, {"Type": "BANK_ROUTING", "BeginOffset": 279, "EndOffset": 289, "Score": 0.999999}], "File": "doc.txt"}
```

## Abrufen der Ergebnisse der Stimmungserkennung

Im Folgenden finden Sie ein Beispiel für eine Ausgabedatei aus einer Analyse, die die in einem Dokument ausgedrückte Stimmung erkannt hat. Sie enthält eine Fehlermeldung, da ein Dokument zu lang ist. Das Format der Eingabe ist ein Dokument pro Zeile. Weitere Informationen finden Sie unter dem Vorgang [DetectSentiment](#).

```
{"File": "50_docs", "Line": 0, "Sentiment": "NEUTRAL", "SentimentScore": {"Mixed": 0.002734508365392685, "Negative": 0.008935936726629734, "Neutral": 0.9841893315315247, "Positive": 0.004140198230743408}} {"File": "50_docs", "Line": 1, "ErrorCode": "DOCUMENT_SIZE_EXCEEDED", "ErrorMessage": "Document size is exceeded maximum size limit 5120 bytes."} {"File": "50_docs", "Line": 2, "Sentiment": "NEUTRAL", "SentimentScore": {"Mixed": 0.0023119584657251835, "Negative": 0.0029857370536774397, "Neutral": 0.9866572022438049, "Positive": 0.008045154623687267}}
```

Im Folgenden finden Sie ein Beispiel für die Ausgabe einer Analyse, bei der das Format der Eingabe ein Dokument pro Datei ist.

```
{"File": "small_doc", "Sentiment": "NEUTRAL", "SentimentScore": {"Mixed": 0.0023450672160834074, "Negative": 0.0009663937962614, "Neutral": 0.9795311689376831, "Positive": 0.017157377675175667}} {"File": "huge_doc", "ErrorCode": "DOCUMENT_SIZE_EXCEEDED", "ErrorMessage": "Document size is exceeds the limit of 5120 bytes."}
```

## Asynchrone Analyse für gezielte Stimmung

Informationen zur Echtzeitanalyse für gezielte Stimmung finden Sie unter [the section called "Echtzeitanalyse für gezielte Stimmung"](#).

Amazon Comprehend bietet die folgenden API-Operationen zum Starten und Verwalten der asynchronen gezielten Stimmungsanalyse:

- [StartTargetedSentimentDetectionJob](#) – Startet einen asynchronen gezielten Stimmungserkennungsauftrag für eine Sammlung von Dokumenten.
- [ListTargetedSentimentDetectionJobs](#) – Gibt die Liste der von Ihnen übermittelten gezielten Stimmungserkennungsaufträge zurück.
- [DescribeTargetedSentimentDetectionJob](#) – Ruft die Eigenschaften (einschließlich Status) ab, die dem angegebenen Zielauftrag zur Stimmungserkennung zugeordnet sind.
- [StopTargetedSentimentDetectionJob](#) – Stoppt den angegebenen in Bearbeitung befindlichen gezielten Stimmungsauftrag.

## Themen

- [Bevor Sie beginnen](#)
- [Analysieren der gezielten Stimmung mithilfe der AWS CLI](#)

## Bevor Sie beginnen

Bevor Sie beginnen, stellen Sie sicher, dass Sie über Folgendes verfügen:

- Eingabe- und Ausgabe-Buckets – Geben Sie die Amazon S3-Buckets an, die Sie für Eingabe und Ausgabe verwenden möchten. Die Buckets müssen sich in derselben -Region befinden wie die API, die Sie aufrufen.
- IAM-Servicerolle – Sie benötigen eine IAM-Servicerolle mit der Berechtigung, auf Ihre Eingabe- und Ausgabe-Buckets zuzugreifen. Weitere Informationen finden Sie unter [Rollenbasierte Berechtigungen, die für asynchrone Operationen erforderlich sind](#).

## Analysieren der gezielten Stimmung mithilfe der AWS CLI

Das folgende Beispiel zeigt die Verwendung der `-StartTargetedSentimentDetectionJobOperation` mit der AWS CLI. In diesem Beispiel wird die Sprache des Eingabetexts angegeben.

Das Beispiel ist für Unix, Linux und macOS formatiert. Ersetzen Sie unter Windows den umgekehrten Schrägstrich (`\`), das Unix-Fortsetzungszeichen, am Ende jeder Zeile durch ein Caret-Zeichen oder Zirkumflex (`^`).

```
aws comprehend start-targeted-sentiment-detection-job \
```

```
--job-name "job name" \  
--language-code "en" \  
--cli-input-json file://path to JSON input file
```

Für den `cli-input-json` Parameter geben Sie den Pfad zu einer JSON-Datei an, die die Anforderungsdaten enthält, wie im folgenden Beispiel gezeigt.

```
{  
  "InputDataConfig": {  
    "S3Uri": "s3://input bucket/input path",  
    "InputFormat": "ONE_DOC_PER_FILE"  
  },  
  "OutputDataConfig": {  
    "S3Uri": "s3://output bucket/output path"  
  },  
  "DataAccessRoleArn": "arn:aws:iam::account ID:role/data access role"  
}
```

Wenn die Anforderung zum Starten des Auftrags erfolgreich war, erhalten Sie die folgende Antwort:

```
{  
  "JobStatus": "SUBMITTED",  
  "JobArn": "job ARN",  
  "JobId": "job ID"  
}
```

## Asynchrone Analyse zur Ereigniserkennung

### Themen

- [Bevor Sie beginnen](#)
- [Erkennen von Ereignissen mithilfe der AWS CLI](#)
- [Auflisten von Ereignissen mit der AWS CLI](#)
- [Beschreiben von Ereignissen mithilfe der AWS CLI](#)
- [Abrufen der Ergebnisse der Ereigniserkennung](#)

Um Ereignisse in einem Dokumentsatz zu erkennen, verwenden Sie die , [StartEventsDetectionJob](#) um einen asynchronen Auftrag zu starten.

## Bevor Sie beginnen

Bevor Sie beginnen, stellen Sie sicher, dass Sie über Folgendes verfügen:

- Eingabe- und Ausgabe-Buckets – Geben Sie die Amazon S3-Buckets an, die Sie für Eingabe und Ausgabe verwenden möchten. Die Buckets müssen sich in derselben -Region befinden wie die API, die Sie aufrufen.
- IAM-Servicerolle – Sie benötigen eine IAM-Servicerolle mit der Berechtigung, auf Ihre Eingabe- und Ausgabe-Buckets zuzugreifen. Weitere Informationen finden Sie unter [Rollenbasierte Berechtigungen, die für asynchrone Operationen erforderlich sind](#).

## Erkennen von Ereignissen mithilfe der AWS CLI

Das folgende Beispiel zeigt die Verwendung der [StartEventsDetectionJob](#) Operation mit der AWS CLI

Das Beispiel ist für Unix, Linux und macOS formatiert. Ersetzen Sie unter Windows den umgekehrten Schrägstrich (\), das Unix-Fortsetzungszeichen, am Ende jeder Zeile durch ein Caret-Zeichen oder Zirkumflex (^).

```
aws comprehend start-events-detection-job \  
  --region region \  
  --job-name job name \  
  --cli-input-json file://path to JSON input file
```

Für den `cli-input-json` Parameter geben Sie den Pfad zu einer JSON-Datei an, die die Anforderungsdaten enthält, wie im folgenden Beispiel gezeigt.

```
{  
  "InputDataConfig": {  
    "S3Uri": "s3://input bucket/input path",  
    "InputFormat": "ONE_DOC_PER_LINE"  
  },  
  "OutputDataConfig": {  
    "S3Uri": "s3://output bucket/output path"  
  },  
  "DataAccessRoleArn": "arn:aws:iam::account ID:role/data access role"  
  "LanguageCode": "en",  
  "TargetEventTypes": [  
    "BANKRUPTCY",  
    "EMPLOYMENT",  
    "CORPORATE_ACQUISITION",
```



```

    "INVESTMENT_GENERAL",
    "CORPORATE_MERGER",
    "IPO",
    "RIGHTS_ISSUE",
    "SECONDARY_OFFERING",
    "SHELF_OFFERING",
    "TENDER_OFFERING",
    "STOCK_SPLIT"
  ]
}

```

Wenn die Anforderung zum Starten des Ereigniserkennungsauftrags erfolgreich war, erhalten Sie die folgende Antwort:

```

{
  "JobStatus": "SUBMITTED",
  "JobId": "job ID"
}

```

## Auflisten von Ereignissen mit der AWS CLI

Verwenden Sie die [-ListEventsDetectionJobs](#) Operation, um eine Liste der von Ihnen übermittelten Aufträge zur Ereigniserkennung anzuzeigen. Die Liste enthält Informationen über die von Ihnen verwendeten Eingabe- und Ausgabespeicherorte sowie den Status der einzelnen Erkennungsaufträge. Das Beispiel ist für Unix, Linux und macOS formatiert. Ersetzen Sie unter Windows den umgekehrten Schrägstrich (\), das Unix-Fortsetzungszeichen, am Ende jeder Zeile durch ein Caret-Zeichen oder Zirkumflex (^).

```
aws comprehend list-events-detection-jobs --region region
```

Als Antwort erhalten Sie JSON ähnlich dem folgenden:

```

{
  "EventsDetectionJobPropertiesList": [
    {
      "DataAccessRoleArn": "arn:aws:iam::account ID:role/data access role",
      "EndTime": timestamp,
      "InputDataConfig": {
        "InputFormat": "ONE_DOC_PER_LINE",
        "S3Uri": "s3://input bucket/input path"
      },
    },
  ],
}

```

```

    "JobId": "job ID",
    "JobName": "job name",
    "JobStatus": "COMPLETED",
    "LanguageCode": "en",
    "Message": "message",
    "OutputDataConfig": {
      "S3Uri": "s3://output bucket/ouput path"
    },
    "SubmitTime": timestamp,
    "TargetEventTypes": [
      "BANKRUPTCY",
      "EMPLOYMENT",
      "CORPORATE_ACQUISITION",
      "INVESTMENT_GENERAL",
      "CORPORATE_MERGER",
      "IPO",
      "RIGHTS_ISSUE",
      "SECONDARY_OFFERING",
      "SHELF_OFFERING",
      "TENDER_OFFERING",
      "STOCK_SPLIT"
    ]
  }
],
"NextToken": "next token"
}

```

## Beschreiben von Ereignissen mithilfe der AWS CLI

Sie können die [DescribeEventsDetectionJob](#) Operation verwenden, um den Status eines vorhandenen Auftrags abzurufen. Das Beispiel ist für Unix, Linux und macOS formatiert. Ersetzen Sie unter Windows den umgekehrten Schrägstrich (\), das Unix-Fortsetzungszeichen, am Ende jeder Zeile durch ein Caret-Zeichen oder Zirkumflex (^).

```

aws comprehend describe-events-detection-job \
  --region region \
  --job-id job ID

```

Als Antwort erhalten Sie das folgende JSON:

```
{
```

```

"EventsDetectionJobProperties": {
  "DataAccessRoleArn": "arn:aws:iam::account ID:role/data access role",
  "EndTime": timestamp,
  "InputDataConfig": {
    "InputFormat": "ONE_DOC_PER_LINE",
    "S3Uri": "S3Uri": "s3://input bucket/input path"
  },
  "JobId": "job ID",
  "JobName": "job name",
  "JobStatus": "job status",
  "LanguageCode": "en",
  "Message": "message",
  "OutputDataConfig": {
    "S3Uri": "s3://output bucket/output path"
  },
  "SubmitTime": timestamp,
  "TargetEventTypes": [
    "BANKRUPTCY",
    "EMPLOYMENT",
    "CORPORATE_ACQUISITION",
    "INVESTMENT_GENERAL",
    "CORPORATE_MERGER",
    "IPO",
    "RIGHTS_ISSUE",
    "SECONDARY_OFFERING",
    "SHELF_OFFERING",
    "TENDER_OFFERING",
    "STOCK_SPLIT"
  ]
}
}

```

## Abrufen der Ergebnisse der Ereigniserkennung

Im Folgenden finden Sie ein Beispiel für eine Ausgabedatei aus einem Analyseauftrag, der Ereignisse in Dokumenten erkannt hat. Das Format der Eingabe ist ein Dokument pro Zeile.

```

{"Entities": [{"Mentions": [{"BeginOffset": 12, "EndOffset": 27, "GroupScore": 1.0, "Score": 0.916355, "Text": "over a year ago", "Type": "DATE"}]}, {"Mentions": [{"BeginOffset": 33, "EndOffset": 39, "GroupScore": 1.0, "Score": 0.996603, "Text": "Amazon", "Type": "ORGANIZATION"}]}, {"Mentions": [{"BeginOffset": 66, "EndOffset": 77, "GroupScore": 1.0, "Score": 0.999283, "Text": "Whole Foods", "Type": "ORGANIZATION"}]}], "Events": [{"Arguments": [{"EntityIndex":

```

```
2, "Role": "INVESTEER", "Score": 0.999283}, {"EntityIndex": 0, "Role": "DATE",
"Score": 0.916355}, {"EntityIndex": 1, "Role": "INVESTOR", "Score": 0.996603}],
"Triggers": [{"BeginOffset": 373, "EndOffset": 380, "GroupScore": 0.999984,
"Score": 0.999955, "Text": "acquire", "Type": "CORPORATE_ACQUISITION"}], "Type":
"CORPORATE_ACQUISITION"}, {"Arguments": [{"EntityIndex": 2, "Role": "PARTICIPANT",
"Score": 0.999283}], "Triggers": [{"BeginOffset": 115, "EndOffset": 123, "GroupScore":
1.0, "Score": 0.999967, "Text": "combined", "Type": "CORPORATE_MERGER"}], "Type":
"CORPORATE_MERGER"}], "File": "doc.txt", "Line": 0}
```

Weitere Informationen zur Struktur der Ereignisausgabedatei und zu unterstützten Ereignistypen finden Sie unter [Ereignisse](#).

## Asynchrone Analyse für die Themenmodellierung

Um die Themen in einem Dokumentsatz zu bestimmen, verwenden Sie die , [StartTopicsDetectionJob](#) um einen asynchronen Auftrag zu starten. Sie können Themen in Dokumenten überwachen, die auf Englisch oder Spanisch geschrieben sind.

### Themen

- [Bevor Sie beginnen](#)
- [Verwendung der AWS Command Line Interface](#)
- [Verwenden des -SDK für Python oder AWS SDK for .NET](#)

### Bevor Sie beginnen

Bevor Sie beginnen, stellen Sie sicher, dass Sie über Folgendes verfügen:

- Eingabe- und Ausgabe-Buckets – Geben Sie die Amazon S3-Buckets an, die Sie für Eingabe und Ausgabe verwenden möchten. Die Buckets müssen sich in derselben -Region befinden wie die API, die Sie aufrufen.
- IAM-Servicerolle – Sie benötigen eine IAM-Servicerolle mit der Berechtigung, auf Ihre Eingabe- und Ausgabe-Buckets zuzugreifen. Weitere Informationen finden Sie unter [Rollenbasierte Berechtigungen, die für asynchrone Operationen erforderlich sind](#).

## Verwendung der AWS Command Line Interface

Das folgende Beispiel zeigt die Verwendung der `-StartTopicsDetectionJobOperation` mit der AWS CLI

Das Beispiel ist für Unix, Linux und macOS formatiert. Ersetzen Sie unter Windows den umgekehrten Schrägstrich (`\`), das Unix-Fortsetzungszeichen, am Ende jeder Zeile durch ein Caret-Zeichen oder Zirkumflex (`^`).

```
aws comprehend start-topics-detection-job \  
    --number-of-topics topics to return \  
    --job-name "job name" \  
    --region region \  
    --cli-input-json file://path to JSON input file
```

Für den `cli-input-json` Parameter geben Sie den Pfad zu einer JSON-Datei an, die die Anforderungsdaten enthält, wie im folgenden Beispiel gezeigt.

```
{  
  "InputDataConfig": {  
    "S3Uri": "s3://input bucket/input path",  
    "InputFormat": "ONE_DOC_PER_FILE"  
  },  
  "OutputDataConfig": {  
    "S3Uri": "s3://output bucket/output path"  
  },  
  "DataAccessRoleArn": "arn:aws:iam::account ID:role/data access role"  
}
```

Wenn die Anforderung zum Starten des Themenerkennungsauftrags erfolgreich war, erhalten Sie die folgende Antwort:

```
{  
  "JobStatus": "SUBMITTED",  
  "JobId": "job ID"  
}
```

Verwenden Sie die [-ListTopicsDetectionJobs](#) Operation, um eine Liste der von Ihnen übermittelten Aufgaben zur Themenerkennung anzuzeigen. Die Liste enthält Informationen über die von Ihnen verwendeten Eingabe- und Ausgabespeicherorte sowie den Status der einzelnen

Erkennungsaufträge. Das Beispiel ist für Unix, Linux und macOS formatiert. Ersetzen Sie unter Windows den umgekehrten Schrägstrich (\), das Unix-Fortsetzungszeichen, am Ende jeder Zeile durch ein Caret-Zeichen oder Zirkumflex (^).

```
aws comprehend list-topics-detection-jobs \-- region
```

Als Antwort erhalten Sie JSON ähnlich dem folgenden:

```
{
  "TopicsDetectionJobPropertiesList": [
    {
      "InputDataConfig": {
        "S3Uri": "s3://input bucket/input path",
        "InputFormat": "ONE_DOC_PER_LINE"
      },
      "NumberOfTopics": topics to return,
      "JobId": "job ID",
      "JobStatus": "COMPLETED",
      "JobName": "job name",
      "SubmitTime": timestamp,
      "OutputDataConfig": {
        "S3Uri": "s3://output bucket/output path"
      },
      "EndTime": timestamp
    },
    {
      "InputDataConfig": {
        "S3Uri": "s3://input bucket/input path",
        "InputFormat": "ONE_DOC_PER_LINE"
      },
      "NumberOfTopics": topics to return,
      "JobId": "job ID",
      "JobStatus": "RUNNING",
      "JobName": "job name",
      "SubmitTime": timestamp,
      "OutputDataConfig": {
        "S3Uri": "s3://output bucket/output path"
      }
    }
  ]
}
```

Sie können die [-DescribeTopicsDetectionJob](#) Operation verwenden, um den Status eines vorhandenen Auftrags abzurufen. Das Beispiel ist für Unix, Linux und macOS formatiert. Ersetzen Sie unter Windows den umgekehrten Schrägstrich (\), das Unix-Fortsetzungszeichen, am Ende jeder Zeile durch ein Caret-Zeichen oder Zirkumflex (^).

```
aws comprehend describe-topics-detection-job --job-id job ID
```

Als Antwort erhalten Sie das folgende JSON:

```
{
  "TopicsDetectionJobProperties": {
    "InputDataConfig": {
      "S3Uri": "s3://input bucket/input path",
      "InputFormat": "ONE_DOC_PER_LINE"
    },
    "NumberOfTopics": topics to return,
    "JobId": "job ID",
    "JobStatus": "COMPLETED",
    "JobName": "job name",
    "SubmitTime": timestamp,
    "OutputDataConfig": {
      "S3Uri": "s3://output bucket/ouput path"
    },
    "EndTime": timestamp
  }
}
```

Verwenden des -SDK für Python oder AWS SDK for .NET

SDK-Beispiele zum Starten eines Themenmodellierungsauftrags finden Sie unter [Verwendung StartTopicsDetectionJob mit einem AWS SDK oder CLI](#).

# Vertrauen und Sicherheit

Benutzer generieren große Mengen an Textinhalten über Online-Anwendungen (wie peer-to-peer Chats und Forum-Diskussionen), Kommentare, die auf Websites veröffentlicht wurden, und über generative KI-Anwendungen (Eingabeaufforderungen und Ausgaben von generativen KI-Modellen). Die Amazon Comprehend Trust and Safety-Funktionen können Ihnen helfen, diese Inhalte zu moderieren, um Ihren Benutzern eine sichere und inklusive Umgebung zu bieten.

Zu den Vorteilen der Vertrauens- und Sicherheitsfunktionen von Amazon Comprehend gehören:

- **Schnellere Moderation:** Moderieren Sie schnell und präzise große Textmengen, um Ihre Online-Plattformen von unangemessenen Inhalten frei zu halten.
- **Anpassbar:** Passen Sie die Moderationsschwellenwerte in API-Antworten an Ihre Anwendungsanforderungen an.
- **Einfach zu bedienen:** Konfigurieren Sie die Vertrauens- und Sicherheitsfunktionen durch LangChain Integration oder mithilfe der AWS CLI oder SDKs .

Amazon Comprehend-Vertrauens- und Sicherheitsmaßnahmen berücksichtigen die folgenden Aspekte der Inhaltsmoderation:

- **Toxicity detection** – Erkennen Sie Inhalte, die schädlich, anstößig oder unangemessen sein können. Beispiele hierfür sind Hasssprache, Bedrohungen oder Missbrauch.
- **Intent classification** – Erkennen Sie Inhalte, die explizite oder implizite böswillige Absichten haben. Beispiele hierfür sind diskriminierende oder illegale Inhalte oder Inhalte, die Empfehlungen zu medizinischen, rechtlichen, politischen, kontroversialen, persönlichen oder finanziellen Themen ausdrücken oder anfordern.
- **Privacy protection** – Benutzer können versehentlich Inhalte bereitstellen, die persönlich identifizierbare Informationen (PII) preisgeben können. Amazon Comprehend PII bietet die Möglichkeit, PII zu erkennen und zu redigieren.

Themen

- [Erkennung der Sichtbarkeit](#)
- [Prompt-Sicherheitsklassifizierung](#)
- [PII-Erkennung und -Schwärzung](#)



# Erkennung der Sichtbarkeit

Die Erkennung von Amazon Comprehend microSD bietet Echtzeiterkennung synchroner Inhalte in textbasierten Interaktionen. Sie können die Erkennung von Trichtern verwenden, um peer-to-peer Konversationen auf Online-Plattformen zu moderieren oder generative KI-Eingaben und -Ausgaben zu überwachen.

Bei der Erkennung der Sichtbarkeit werden die folgenden Kategorien anstößiger Inhalte erkannt:

## GRAPHIC

Grafische Sprache verwendet visuell beschreibende, detaillierte und geografisch anschauliche Bilder. Eine solche Sprache wird oft ausführlich dargestellt, um einen Beleidigungsversuch, eine Anfeindung oder einen Gesundheitsvorfall für den Empfänger auszunutzen.

## HARASSMENT\_OR\_ABUSE

Sprache, die unabhängig von der Absicht eine störende Stromentwicklung zwischen dem Sprecher und dem Sprecher auslöst, versucht, sich auf das Widerstandsfähigkeit des Empfängers auszuwirken oder eine Person zu objektieren.

## HATE\_SPEECH

Sprache, die eine Person oder eine Gruppe auf der Grundlage einer Identität beleidigt, beleidigt, beleidigt oder entmenschlich macht, sei es E-Mail-Abstammung, Benachteiligung, Geschlecht, Geschlecht, Orientierung, Fähigkeit, nationaler Ursprung oder eine andere Identitätsgruppe.

## INSULT

Sprache, die erniedrigende, erniedrigende, simulierende, beleidigende oder herabsetzende Sprache umfasst.

## PROFANITY

Sprache, die Wörter, Wortgruppen oder Akronyme enthält, die unhöflich, anstößig oder anstößig sind, wird als unecht angesehen.

## SEXUAL

Sprache, die auf ein gewisses Interesse, eine Aktivität oder einen erregenden Hinweis hinweist, indem direkte oder indirekte Verweise auf Körperteile oder physische Merkmale oder Ker verwendet werden.

## VIOLENCE\_OR\_THREAT

Sprache, die Bedrohungen umfasst, die einer Person oder Gruppe schädliche, benachteiligte oder feindselige Aktivitäten zuzufügen versuchen.

## TOXICITY

Sprache, die Wörter, Wortgruppen oder Akronyme enthält, die in einer der oben genannten Kategorien als statisch angesehen werden könnten.

## Erkennen schädlicher Inhalte mithilfe der API

Verwenden Sie die synchrone [-DetectToxicContent](#) Operation, um schädliche Inhalte im Text zu erkennen. Dieser Vorgang führt eine Analyse für eine Liste von Textzeichenfolgen durch, die Sie als Eingabe angeben. Die API-Antwort enthält eine Ergebnisliste, die der Größe der Eingabeliste entspricht.

Derzeit unterstützt die Erkennung schädlicher Inhalte nur die englische Sprache. Für Eingabetext können Sie eine Liste mit bis zu 10 Textzeichenfolgen angeben. Jede Zeichenfolge hat eine maximale Größe von 1KB.

Die Erkennung schädlicher Inhalte gibt eine Liste der Analyseergebnisse zurück, einen Eintrag in der Liste für jede Eingabezeichenfolge. Ein Eintrag enthält eine Liste der in der Textzeichenfolge identifizierten Inhaltstypen sowie einen Konfidenzwert für jeden Inhaltstyp. Der Eintrag enthält auch einen Farbwert für die Zeichenfolge.

Die folgenden Beispiele zeigen, wie Sie die `-DetectToxicContent` Operation mit und AWS CLI Python verwenden.

### AWS CLI

Sie können schädliche Inhalte mit dem folgenden Befehl in der erkennen AWS CLI:

```
aws comprehend detect-toxic-content --language-code en /  
--text-segments "[{"Text\":"You are so obtuse\"}]"
```

Der AWS CLI antwortet mit dem folgenden Ergebnis. Das Textsegment erhält einen hohen Konfidenzwert in der INSULT Kategorie, mit einem daraus resultierenden hohen Kompressionswert:

```
{
```

```

"ResultList": [
  {
    "Labels": [
      {
        "Name": "PROFANITY",
        "Score": 0.0006000000284984708
      },
      {
        "Name": "HATE_SPEECH",
        "Score": 0.00930000003427267
      },
      {
        "Name": "INSULT",
        "Score": 0.9204999804496765
      },
      {
        "Name": "GRAPHIC",
        "Score": 9.99999747378752e-05
      },
      {
        "Name": "HARASSMENT_OR_ABUSE",
        "Score": 0.0052999998442828655
      },
      {
        "Name": "SEXUAL",
        "Score": 0.01549999974668026
      },
      {
        "Name": "VIOLENCE_OR_THREAT",
        "Score": 0.007799999788403511
      }
    ],
    "Toxicity": 0.7192999720573425
  }
]
}

```

Sie können bis zu 10 Textzeichenfolgen im folgenden Format für den `text-segments` Parameter eingeben:

```

--text-segments "[{"Text\":"text string 1\"},
                  {"Text\":"text string2\"},

```

```
{\"Text\": \"text string\"}]
```

Der AWS CLI antwortet mit den folgenden Ergebnissen:

```
{
  "ResultList": [
    {
      "Labels": [ (truncated) ],
      "Toxicity": 0.3192999720573425
    },
    {
      "Labels": [ (truncated) ],
      "Toxicity": 0.1192999720573425
    },
    {
      "Labels": [ (truncated) ],
      "Toxicity": 0.0192999720573425
    }
  ]
}
```

## Python (Boto)

Das folgende Beispiel zeigt, wie Sie schädliche Inhalte mit Python erkennen:

```
import boto3
client = boto3.client(
    service_name='comprehend',
    region_name=region) # For example, 'us-west-2'

response = client.detect_toxic_content(
    LanguageCode='en',
    TextSegments=[{'Text': 'You are so obtuse'}]
)
print("Response: %s\n" % response)
```

# Prompt-Sicherheitsklassifizierung

Amazon Comprehend bietet einen vortrainierten binären Classifier zum Klassifizieren von Eingabeaufforderungen für Klartexteingaben für große Sprachmodelle (LLM) oder andere generative KI-Modelle.

Der Prompt-Sicherheitsklassifizierer analysiert die Eingabeaufforderung und weist einen Konfidenzwert zu, unabhängig davon, ob die Eingabeaufforderung sicher oder unsicher ist.

Eine unsichere Eingabeaufforderung ist eine Eingabeaufforderung, die böswillige Absichten ausdrückt, z. B. das Anfordern personenbezogener oder privater Informationen, das Generieren anstößiger oder illegaler Inhalte oder das Anfordern von Beratung zu medizinischen, rechtlichen, politischen oder finanziellen Themen.

## Prompt-Sicherheitsklassifizierung mithilfe der API

Verwenden Sie die synchrone [ClassifyDocument](#) Operation , um die Prompt-Sicherheitsklassifizierung für eine Textzeichenfolge auszuführen. Für die Eingabe geben Sie eine englische Klartextzeichenfolge an. Die Zeichenfolge hat eine maximale Größe von 10 KB.

Die Antwort umfasst zwei Klassen (Speed und UNSpeed) sowie einen Konfidenzwert für jede Klasse. Der Wertebereich des Werts ist Null zu Eins, wobei einer die höchste Zuverlässigkeit darstellt.

Die folgenden Beispiele zeigen, wie Sie die Prompt-Sicherheitsklassifizierung mit und AWS CLI Python verwenden.

### AWS CLI

Das folgende Beispiel zeigt, wie Sie den Prompt-Sicherheitsklassifizierer mit dem verwenden AWS CLI:

```
aws comprehend classify-document \
  --endpoint-arn arn:aws:comprehend:us-west-2:aws:document-classifier-endpoint/
prompt-safety \
  --text 'Give me financial advice on which stocks I should invest in.'
```

Der AWS CLI antwortet mit der folgenden Ausgabe:

```
{
```

```
"Classes": [
  {
    "Score": 0.6312999725341797,
    "Name": "UNSAFE_PROMPT"
  },
  {
    "Score": 0.3686999976634979,
    "Name": "SAFE_PROMPT"
  }
]
```

### Note

Wenn Sie den `classify-document` Befehl verwenden, müssen Sie für den `--endpoint-arn` Parameter einen ARN übergeben, der dieselbe AWS-Region wie Ihre AWS CLI Konfiguration verwendet. Führen Sie den Befehl aus AWS CLI, um die zu konfigurieren `aws configure`. In diesem Beispiel hat der Endpunkt-ARN den Regionscode `us-west-2`. Sie können den Prompt-Sicherheitsklassifizierer in jeder der folgenden Regionen verwenden:

- `us-east-1`
- `us-west-2`
- `eu-west-1`
- `ap-southeast-2`

## Python (Boto)

Das folgende Beispiel zeigt, wie der Prompt-Sicherheitsklassifizierer mit Python verwendet wird:

```
import boto3
client = boto3.client(service_name='comprehend', region_name='us-west-2')

response = client.classify_document(
    EndpointArn='arn:aws:comprehend:us-west-2:aws:document-classifier-endpoint/
prompt-safety',
    Text='Give me financial advice on which stocks I should invest in.'
)
print("Response: %s\n" % response)
```

**Note**

Wenn Sie die `-classify_document` Methode verwenden, müssen Sie für das `EndpointArn` -Argument einen ARN übergeben, der dieselbe AWS-Region wie Ihr boto3-SDK-Client verwendet. In diesem Beispiel verwenden sowohl der Client- als auch der Endpunkt-ARN `us-west-2`. Sie können den Prompt-Sicherheitsklassifizierer in jeder der folgenden Regionen verwenden:

- `us-east-1`
- `us-west-2`
- `eu-west-1`
- `ap-southeast-2`

## PII-Erkennung und -Schwärzung

Sie können die Amazon Comprehend-Konsole oder APIs verwenden, um persönlich identifizierbare Informationen (PII) in englischen oder spanischen Textdokumenten zu erkennen. PII ist ein Textverweis auf personenbezogene Daten, die eine Person identifizieren können. Beispiele für PII sind Adressen, Bankkontonummern und Telefonnummern.

Sie können die PII-Entitäten im Text erkennen oder redigieren. Um PII-Entitäten zu erkennen, können Sie Echtzeitanalysen oder einen asynchronen Batch-Auftrag verwenden. Um die PII-Entitäten zu redigieren, müssen Sie einen asynchronen Batch-Auftrag verwenden.

Weitere Informationen finden Sie unter [Persönlich Identifizierbare Informationen \(PII\)](#).

# Persönlich Identifizierbare Informationen (PII)

Sie können die Amazon Comprehend Comprehend-Konsole oder APIs verwenden, um personenbezogene Daten (PII) in englischen oder spanischen Textdokumenten zu erkennen. PII ist ein Textverweis auf personenbezogene Daten, der zur Identifizierung einer Person verwendet werden könnte. Zu den Beispielen für personenbezogene Daten gehören Adressen, Bankkontonummern und Telefonnummern.

Bei der PII-Erkennung haben Sie die Wahl, ob Sie die PII-Entitäten lokalisieren oder die PII-Entitäten im Text redigieren möchten. Um PII-Entitäten zu lokalisieren, können Sie eine Echtzeitanalyse oder einen asynchronen Batch-Job verwenden. Um die PII-Entitäten zu redigieren, müssen Sie einen asynchronen Batch-Job verwenden.

Sie können Amazon S3 Object Lambda Access Points für persönlich identifizierbare Informationen (PII) verwenden, um den Abruf von Dokumenten aus Ihrem Amazon S3 S3-Bucket zu steuern. Sie können den Zugriff auf Dokumente kontrollieren, die personenbezogene Daten enthalten, und personenbezogene Daten aus den Dokumenten unkenntlich machen. Weitere Informationen finden Sie unter [Verwenden von Amazon S3-Objekt-Lambda-Zugriffspunkten für persönlich identifizierbare Informationen \(PII\)](#).

## Themen

- [Erkennung von PII-Entitäten](#)
- [Kennzeichnung von PII-Entitäten](#)
- [PII-Echtzeitanalyse \(Konsole\)](#)
- [Asynchrone PII-Analyseaufträge \(Konsole\)](#)
- [PII-Echtzeitanalyse \(API\)](#)
- [Asynchrone PII-Analyseaufträge \(API\)](#)

## Erkennung von PII-Entitäten

Sie können Amazon Comprehend verwenden, um PII-Entitäten in englischen oder spanischen Textdokumenten zu erkennen. Eine PII-Entität ist eine bestimmte Art von persönlich identifizierbaren Informationen (PII). Verwenden Sie die PII-Erkennung, um die PII-Entitäten zu lokalisieren oder die PII-Entitäten im Text zu schwärzen.

## Themen



- [Suchen Sie nach PII-Entitäten](#)
- [PII-Entitäten redigieren](#)
- [Universelle PII-Entitätstypen](#)
- [Länderspezifische PII-Entitätstypen](#)

## Suchen Sie nach PII-Entitäten

Um die PII-Entitäten in Ihrem Text zu finden, können Sie mithilfe von Echtzeitanalysen schnell ein einzelnes Dokument analysieren. Sie können auch einen asynchronen Batch-Job für eine Sammlung von Dokumenten starten.

Sie können die Konsole oder die API für die Echtzeitanalyse eines einzelnen Dokuments verwenden. Ihr Eingabetext kann bis zu 100 Kilobyte an UTF-8-kodierten Zeichen enthalten.

Sie können beispielsweise den folgenden Eingabetext einreichen, um die PII-Entitäten zu finden:

Hallo Paulo Santos. Der letzte Kontoauszug für Ihr Kreditkartenkonto 1111-0000-1111-0000 wurde an 123 Any Street, Seattle, WA 98109 gesendet.

Die Ausgabe enthält die Information, dass „Paul Santos“ den Typ hat, „1111-0000-1111-0000“ den Typ hat und „123 Any StreetNAME, Seattle, WA 98109“ den Typ hat. CREDIT\_DEBIT\_NUMBER ADDRESS

Amazon Comprehend gibt eine Liste der erkannten PII-Entitäten mit den folgenden Informationen für jede PII-Entität zurück:

- Ein Wert, der die Wahrscheinlichkeit schätzt, dass es sich bei der erkannten Textspanne um den erkannten Entitätstyp handelt.
- Der PII-Entitätstyp.
- Die Position der PII-Entität im Dokument, angegeben als Zeichenversätze für den Anfang und das Ende der Entität.

Beispielsweise erzeugt der zuvor erwähnte Eingabetext die folgende Antwort:

```
{
  "Entities": [
    {
```

```

        "Score": 0.9999669790267944,
        "Type": "NAME",
        "BeginOffset": 6,
        "EndOffset": 18
    },
    {
        "Score": 0.8905550241470337,
        "Type": "CREDIT_DEBIT_NUMBER",
        "BeginOffset": 69,
        "EndOffset": 88
    },
    {
        "Score": 0.9999889731407166,
        "Type": "ADDRESS",
        "BeginOffset": 103,
        "EndOffset": 138
    }
]
}

```

## PII-Entitäten redigieren

Um die PII-Entitäten in Ihrem Text zu schwärzen, können Sie die Konsole oder die API verwenden, um einen asynchronen Batch-Job zu starten. Amazon Comprehend gibt eine Kopie des eingegebenen Textes mit Schwärzungen für jede PII-Entität zurück.

Sie können beispielsweise den folgenden Eingabetext einreichen, um die PII-Entitäten zu redigieren:

Hallo Paulo Santos. Der letzte Kontoauszug für Ihr Kreditkartenkonto 1111-0000-1111-0000 wurde an 123 Any Street, Seattle, WA 98109 gesendet.

Die Ausgabedatei enthält den folgenden Text:

Hallo \*\*\*\*\*. Die letzte Abrechnung für Ihr Kreditkartenkonto \*\*\*\*\* wurde per Post an \*\*\* \*\* \*\*\*\*\* \*\* \*\*\*\*\* gesendet.

## Universelle PII-Entitätstypen

Einige PII-Entitätstypen sind universal (nicht spezifisch für einzelne Länder), wie E-Mail-Adressen und Kreditkartennummern. Amazon Comprehend erkennt die folgenden Typen von universellen PII-Entitäten:

## ADDRESS

Eine physische Adresse, z. B. „100 Main Street, Anytown, USA“ oder „Suite #12, Building 123“. Eine Adresse kann Informationen wie Straße, Gebäude, Standort, Stadt, Bundesland, Land, Bezirk, Postleitzahl, Bezirk und Nachbarschaft enthalten.

## AGE

Das Alter einer Person, einschließlich Menge und Zeiteinheit. In der Phrase „Ich bin 40 Jahre alt“ erkennt Amazon Comprehend beispielsweise „40 Jahre“ als Alter an.

## AWS\_ACCESS\_KEY

Eine eindeutige Kennung, die einem geheimen Zugriffsschlüssel zugeordnet ist. Sie verwenden die Zugriffsschlüssel-ID und den geheimen Zugriffsschlüssel, um programmatische AWS Anfragen kryptografisch zu signieren.

## AWS\_SECRET\_KEY

Eine eindeutige Kennung, die einem Zugriffsschlüssel zugeordnet ist. Sie verwenden die Zugriffsschlüssel-ID und den geheimen Zugriffsschlüssel, um programmatische AWS Anfragen kryptografisch zu signieren.

## CREDIT\_DEBIT\_CVV

Ein dreistelliger Kartenbestätigungscode (CVV), der auf Kredit- und Debitkarten von VISA und Discover vorhanden ist. MasterCard Bei Kredit- oder Debitkarten von American Express ist der CVV ein vierstelliger numerischer Code.

## CREDIT\_DEBIT\_EXPIRATION

Das Ablaufdatum einer Kredit- oder Debitkarte. Diese Zahl ist normalerweise vierstellig und wird oft als Monat/Jahr oder MM/JJ formatiert. Amazon Comprehend erkennt Ablaufdaten wie den 21. Januar, den Januar 2021 und den Januar 2021.

## CREDIT\_DEBIT\_NUMBER

Die Nummer einer Kredit- oder Debitkarte. Diese Zahlen können zwischen 13 und 16 Ziffern lang sein. Amazon Comprehend erkennt jedoch auch Kredit- oder Debitkartennummern, wenn nur die letzten vier Ziffern vorhanden sind.

## DATE\_TIME

Ein Datum kann ein Jahr, einen Monat, einen Tag, einen Wochentag oder eine Tageszeit enthalten. Amazon Comprehend erkennt beispielsweise „19. Januar 2020“ oder „11 Uhr“ als

Daten. Amazon Comprehend erkennt Teildaten, Datumsbereiche und Datumsintervalle. Es erkennt auch Jahrzehnte wie „die 1990er“.

## TREIBER\_ID

Die Nummer, die einem Führerschein zugewiesen ist. Dabei handelt es sich um ein offizielles Dokument, das es einer Person ermöglicht, ein oder mehrere motorisierte Fahrzeuge auf einer öffentlichen Straße zu fahren. Eine Führerscheinnummer besteht aus alphanumerischen Zeichen.

## EMAIL

Eine E-Mail-Adresse, z. B. marymajor@email.com.

## INTERNATIONALE\_BANKKONTONUMMER

Eine internationale Bankkontonummer hat in jedem Land spezifische Formate. Weitere Informationen finden Sie [unter www.iban.com/structure](http://www.iban.com/structure).

## IP\_ADDRESS

Eine IPv4-Adresse, z. B. 198.51.100.0.

## LICENSE\_PLATE

Ein Nummernschild für ein Fahrzeug wird von dem Staat oder Land ausgestellt, in dem das Fahrzeug registriert ist. Das Format für Personenkraftwagen besteht in der Regel aus fünf bis acht Ziffern, die aus Großbuchstaben und Zahlen bestehen. Das Format variiert je nach Standort des ausstellenden Staates oder Landes.

## MAC\_ADRESS

Eine MAC-Adresse (Media Access Control) ist eine eindeutige Kennung, die einem Network Interface Controller (NIC) zugewiesen wird.

## NAME

Der Name einer Person. Dieser Entitätstyp umfasst keine Titel wie Dr., Herr, Frau oder Fräulein. Amazon Comprehend wendet diesen Entitätstyp nicht auf Namen an, die Teil von Organisationen oder Adressen sind. Amazon Comprehend erkennt beispielsweise die „John Doe Organization“ als Organisation und „Jane Doe Street“ als Adresse an.

## PASSWORD

Eine alphanumerische Zeichenfolge, die als Passwort verwendet wird, z. B. „\*very20special #pass\*“.

## PHONE

Eine Telefonnummer. Dieser Entitätstyp umfasst auch Fax- und Pager-Nummern.

## PIN

Eine vierstellige persönliche Identifikationsnummer (PIN), mit der Sie auf Ihr Bankkonto zugreifen können.

## SWIFT\_CODE

Ein SWIFT-Code ist ein Standardformat für den Bank Identifier Code (BIC), der zur Angabe einer bestimmten Bank oder Filiale verwendet wird. Banken verwenden diese Codes für Geldtransfers wie internationale Überweisungen.

SWIFT-Codes bestehen aus acht oder 11 Zeichen. Die 11-stelligen Codes beziehen sich auf bestimmte Filialen, während sich achtstellige Codes (oder 11-stellige Codes, die auf „XXX“ enden) auf den Hauptsitz oder die Hauptniederlassung beziehen.

## URL

Eine Webadresse, z. B. [www.example.com](http://www.example.com).

Die Datei „snowball-adapter.config“ enthält die Konfigurationseinstellungen für den Adapter.

Ein Benutzername, der ein Konto identifiziert, z. B. ein Anmeldename, ein Bildschirmname, ein Spitzname oder ein Nutzernamen.

## FAHRZEUGIDENTIFIKATIONSNUMMER

Eine Fahrzeugidentifikationsnummer (VIN) identifiziert ein Fahrzeug eindeutig. Inhalt und Format der Fahrgestellnummer sind in der Spezifikation ISO 3779 definiert. Jedes Land hat spezifische Codes und Formate für VINs.

## Länderspezifische PII-Entitätstypen

Einige PII-Entitätstypen sind länderspezifisch, z. B. Passnummern und andere von der Regierung ausgestellte Ausweisnummern. Amazon Comprehend erkennt die folgenden Typen von länderspezifischen PII-Entitäten:

### CA\_HEALTH\_NUMBER

Eine kanadische Gesundheitsdienstnummer ist eine 10-stellige eindeutige Kennung, die für den Zugang zu Gesundheitsleistungen erforderlich ist.

## CA\_SOCIAL\_INSURANCE\_NUMBER

Eine kanadische Sozialversicherungsnummer (SIN) ist eine neunstellige eindeutige Kennung, die Einzelpersonen benötigen, um staatliche Programme und Leistungen in Anspruch zu nehmen.

Die SIN ist als drei Gruppen von drei Ziffern formatiert, z. B. 123-456-789. [Eine SIN kann durch ein einfaches Prüfziffernverfahren, den so genannten Luhn-Algorithmus, validiert werden.](#)

## IN\_AADHAAR

Ein indisches Aadhaar ist eine 12-stellige eindeutige Identifikationsnummer, die von der indischen Regierung an Einwohner Indiens vergeben wird. Das Aadhaar-Format hat nach der vierten und achten Ziffer ein Leerzeichen oder einen Bindestrich.

## IN\_NREGA

Eine Nummer nach dem Indian National Rural Employment Guarantee Act (NREGA) besteht aus zwei Buchstaben, gefolgt von 14 Zahlen.

## IN\_PERMANENT\_ACCOUNT\_NUMBER

Eine indische permanente Kontonummer ist eine 10-stellige eindeutige alphanumerische Nummer, die von der Einkommensteuerbehörde ausgestellt wird.

## IN\_VOTER\_NUMBER

Ein indischer Wählerausweis besteht aus drei Buchstaben, gefolgt von sieben Zahlen.

## UK\_NATIONAL\_HEALTH\_SERVICE\_NUMBER

Eine Nummer des britischen National Health Service ist eine 10- bis 17-stellige Nummer, z. B. 485 777 3456. Das aktuelle System formatiert die 10-stellige Zahl mit Leerzeichen nach der dritten und sechsten Ziffer. Die letzte Ziffer ist eine Prüfsumme zur Fehlererkennung.

Das 17-stellige Zahlenformat enthält Leerzeichen nach der 10. und 13. Ziffer.

## UK\_NATIONAL\_INSURANCE\_NUMBER

Eine britische National Insurance Number (NINO) ermöglicht Einzelpersonen den Zugang zu Leistungen der nationalen Versicherung (Sozialversicherung). Sie wird auch für einige Zwecke im britischen Steuersystem verwendet.

Die Zahl ist neunstellig und beginnt mit zwei Buchstaben, gefolgt von sechs Zahlen und einem Buchstaben. Ein NINO kann mit einem Leerzeichen oder einem Gedankenstrich nach den beiden Buchstaben und nach der zweiten, vierten und sechsten Ziffer formatiert werden.

## UK\_UNIQUE\_TAXPAYER\_REFERENCE\_NUMBER

Eine Unique Taxpayer Reference (UTR) für Großbritannien ist eine 10-stellige Zahl, die einen Steuerzahler oder ein Unternehmen identifiziert.

## BANKKONTONUMMER

Eine US-Bankkontonummer, die in der Regel 10 bis 12 Ziffern lang ist. Amazon Comprehend erkennt auch Bankkontonummern, wenn nur die letzten vier Ziffern vorhanden sind.

## BANK\_ROUTING

Eine US-Bankleitzahl. Diese sind normalerweise neun Ziffern lang, aber Amazon Comprehend erkennt auch Routing-Nummern, wenn nur die letzten vier Ziffern vorhanden sind.

## REISEPASSNUMMER

Eine US-Passnummer. Die Passnummern bestehen aus sechs bis neun alphanumerischen Zeichen.

## US\_INDIVIDUAL\_TAX\_IDENTIFICATION\_NUMBER

Eine US-amerikanische Steueridentifikationsnummer (ITIN) ist eine neunstellige Zahl, die mit einer „9“ beginnt und eine „7“ oder „8“ als vierte Ziffer enthält. Eine ITIN kann mit einem Leerzeichen oder einem Gedankenstrich nach der dritten und vierten Ziffer formatiert werden.

## SSN

Eine US-Sozialversicherungsnummer (SSN) ist eine neunstellige Zahl, die an US-Bürger, ständige Einwohner und vorübergehend erwerbstätige Einwohner vergeben wird. Amazon Comprehend erkennt auch Sozialversicherungsnummern, wenn nur die letzten vier Ziffern vorhanden sind.

## Kennzeichnung von PII-Entitäten

Wenn Sie die PII-Erkennung ausführen, gibt Amazon Comprehend die Labels der identifizierten PII-Entitätstypen zurück. Wenn Sie beispielsweise den folgenden Eingabetext an Amazon Comprehend senden:

Hallo Paulo Santos. Der letzte Kontoauszug für Ihr Kreditkartenkonto 1111-0000-1111-0000 wurde an 123 Any Street, Seattle, WA 98109 gesendet.

Die Ausgabe umfasst Beschriftungen, die PII-Entitätstypen repräsentieren, zusammen mit einem Konfidenzwert für die Genauigkeit. In diesem Fall generiert der Dokumenttext „Paul Santos“,

„1111-0000-1111-0000“ und „123 Any Street, Seattle, WA 98109“ die Beschriftungen bzw. als PII-Entitätstypen. NAME CREDIT\_DEBIT\_NUMBER ADDRESS Weitere Informationen [Universelle PII-Entitätstypen](#) zu unterstützten Entitätstypen finden Sie unter.

Amazon Comprehend bietet die folgenden Informationen für jedes Etikett:

- Der Labelname des PII-Entitätstyps.
- Ein Wert, der die Wahrscheinlichkeit schätzt, dass der erkannte Text als PII-Entitätstyp gekennzeichnet ist.

Das obige Beispiel für einen Eingabetext führt zu der folgenden JSON-Ausgabe.

```
{
  "Labels": [
    {
      "Name": "NAME",
      "Score": 0.9149109721183777
    },
    {
      "Name": "CREDIT_DEBIT_NUMBER",
      "Score": 0.5698626637458801
    }
  ],
  {
    "Name": "ADDRESS",
    "Score": 0.9951046109199524
  }
  ]
}
```

## PII-Echtzeitanalyse (Konsole)

Sie können die Konsole verwenden, um die PII-Echtzeiterkennung eines Textdokuments auszuführen. Die maximale Textgröße beträgt 100 Kilobyte an UTF-8-kodierten Zeichen. In der Konsole werden die Ergebnisse angezeigt, sodass Sie die Analyse überprüfen können.

Führen Sie mithilfe des integrierten Modells eine Echtzeitanalyse zur Erkennung personenbezogener Daten durch

1. [Melden Sie sich bei der Amazon Comprehend Comprehend-Konsole an AWS Management Console und öffnen Sie sie unter https://console.aws.amazon.com/comprehend/](https://console.aws.amazon.com/comprehend/)



2. Wählen Sie im linken Menü die Option Echtzeitanalyse aus.
3. Wählen Sie unter Eingabetyp die Option Integriert als Analysetyp aus.
4. Geben Sie den Text ein, den Sie analysieren möchten.
5. Wählen Sie Analysieren. Die Konsole zeigt die Ergebnisse der Textanalyse im Bereich Einblicke an. Auf der Registerkarte „PII“ werden die PII-Entitäten aufgeführt, die in Ihrem Eingabetext erkannt wurden.

Im Bereich „Einblicke“ werden auf der Registerkarte „PII“ Ergebnisse für zwei Analysemodi angezeigt:

- Offsets — identifiziert die Position von PII im Textdokument.
- Labels — identifiziert die Labels der identifizierten PII-Entitätstypen.

## Offsets

Der Analysemodus Offsets identifiziert die Position personenbezogener Daten in Ihren Textdokumenten. Weitere Informationen finden Sie unter [Suchen Sie nach PII-Entitäten](#).

**Insights** Info

---

Entities
Key phrases
Language
PII
Sentiment
Targeted sentiment
Syntax

Personally identifiable information (PII) analysis mode

Offsets  
Identify the location of PII in your text documents.

Labels  
Label text documents with PII.

**Analyzed text**

Hello [Zhang Wei](#), I am [John](#). Your AnyCompany Financial Services, LLC credit card account [1111-0000-1111-0008](#) has a minimum payment of \$24.53 that is due by [July 31st](#). Based on your autopay settings, we will withdraw your payment on the due date from your bank account number [XXXXXX1111](#) with the routing number [XXXXX0000](#).

Customer feedback for Sunshine Spa, [123 Main St](#), Anywhere. Send comments to [Alice](#) at [sunspa@mail.com](mailto:sunspa@mail.com).

I enjoyed visiting the spa. It was very comfortable but it was also very expensive. The amenities were ok but the service made the spa a great experience.

**▼ Results**

< 1 > ⚙️

Entity	Type	Confidence
Zhang Wei	Name	0.99+
John	Name	0.99+
1111-0000-1111-0008	Credit debit number	0.99+
July 31st	Date time	0.99+
XXXXXX1111	Bank account number	0.99+
XXXXX0000	Bank routing	0.99+
123 Main St	Address	0.99+
Alice	Name	0.99+
sunspa@mail.com	Email	0.99+

**▶ Application integration**

## Labels

Der Analysemodus Labels gibt die Labels der identifizierten PII-Entitätstypen zurück. Weitere Informationen finden Sie unter [Kennzeichnung von PII-Entitäten](#).

The screenshot displays the 'Insights Info' section of the Amazon Comprehend console. The 'PII' tab is selected, showing 'Personally identifiable information (PII) analysis mode'. Under 'Offsets', the 'Labels' option is selected. The 'Results' section shows a table of detected PII entities with their types and confidence scores.

Type	Confidence
Email	1.00
Name	0.99+
Bank account number	0.98
Bank routing	0.69
Credit debit number	1.00

## Asynchrone PII-Analyseaufträge (Konsole)

Sie können die Konsole verwenden, um asynchrone Analysejobs zur Erkennung von PII-Entitäten zu erstellen. Weitere Informationen zu PII-Entitätstypen finden Sie unter [Erkennung von PII-Entitäten](#)

Um einen Analysejob zu erstellen

1. [Melden Sie sich bei der Amazon Comprehend Comprehend-Konsole an AWS Management Console und öffnen Sie sie unter https://console.aws.amazon.com/comprehend/](https://console.aws.amazon.com/comprehend/)
2. Wählen Sie im linken Menü Analyse-Jobs und dann Create Job aus.
3. Geben Sie dem Analysejob unter Auftragseinstellungen einen eindeutigen Namen.
4. Wählen Sie als Analysetyp die Option Persönlich identifizierbare Informationen (PII) aus.
5. Wählen Sie unter Sprache eine der unterstützten Sprachen (Englisch oder Spanisch) aus.
6. Wählen Sie im Ausgabemodus eine der folgenden Optionen aus:

- Offsets — Die Jobausgabe gibt den Standort jeder PII-Entität zurück.
  - Schwärzungen — Die Jobausgabe gibt eine Kopie des Eingabetextes zurück, wobei jeder PII-Eintrag geschwärzt wurde.
7. (Optional) Wenn Sie Schwärzungen als Ausgabemodus wählen, können Sie die PII-Entitätstypen auswählen, die geschwärzt werden sollen.
  8. Geben Sie unter Eingabedaten an, wo sich die Eingabedokumente in Amazon S3 befinden:
    - Um Ihre eigenen Dokumente zu analysieren, wählen Sie Meine Dokumente und dann Browse S3, um den Pfad zu dem Bucket oder Ordner anzugeben, der Ihre Dateien enthält.
    - Um die von Amazon Comprehend bereitgestellten Proben zu analysieren, wählen Sie Beispieldokumente aus. In diesem Fall verwendet Amazon Comprehend einen Bucket, der von verwaltet wird AWS, und Sie geben den Standort nicht an.
  9. (Optional) Geben Sie als Eingabeformat eines der folgenden Formate für Ihre Eingabedateien an:
    - Ein Dokument pro Datei — Jede Datei enthält ein Eingabedokument. Dies eignet sich am besten für Sammlungen großer Dokumente.
    - Ein Dokument pro Zeile — Die Eingabe besteht aus einer oder mehreren Dateien. Jede Zeile in einer Datei wird als Dokument betrachtet. Dies eignet sich am besten für kurze Dokumente wie Beiträge in sozialen Netzwerken. Jede Zeile muss mit einem Zeilenvorschub (LF,\n), einem Zeilenwechsel (CR,\r) oder beidem (CRLF,\r\n) enden. Sie können das UTF-8-Zeilentrennzeichen (u+2028) nicht verwenden, um eine Zeile zu beenden.
  10. Wählen Sie unter Ausgabedaten die Option Browse S3 aus. Wählen Sie den Amazon S3 S3-Bucket oder -Ordner aus, in den Amazon Comprehend die durch die Analyse erzeugten Ausgabedaten schreiben soll.
  11. (Optional) Um das Ausgabeergebnis Ihres Jobs zu verschlüsseln, wählen Sie Verschlüsselung. Wählen Sie dann aus, ob Sie einen KMS-Schlüssel verwenden möchten, der mit dem aktuellen Konto verknüpft ist, oder einen von einem anderen Konto:
    - Wenn Sie einen Schlüssel verwenden, der dem aktuellen Konto zugeordnet ist, wählen Sie den Schlüsselalias oder die Schlüssel-ID für die KMS-Schlüssel-ID.
    - Wenn Sie einen Schlüssel verwenden, der einem anderen Konto zugeordnet ist, geben Sie den ARN für den Schlüsselalias oder die Schlüssel-ID unter KMS-Schlüssel-ID ein.

**Note**

Weitere Informationen zur Erstellung und Verwendung von KMS-Schlüsseln und der zugehörigen Verschlüsselung finden Sie unter [Schlüsselverwaltungsdienst \(KMS\)](#).

12. Geben Sie unter Zugriffsberechtigungen eine IAM-Rolle an, die:

- Gewährt Lesezugriff auf den Amazon S3 S3-Speicherort Ihrer Eingabedokumente.
- Gewährt Schreibzugriff auf den Amazon S3 S3-Speicherort Ihrer Ausgabedokumente.
- Beinhaltet eine Vertrauensrichtlinie, die es dem `comprehend.amazonaws.com` Service Principal ermöglicht, die Rolle zu übernehmen und seine Berechtigungen zu erhalten.

Wenn Sie noch keine IAM-Rolle mit diesen Berechtigungen und einer entsprechenden Vertrauensrichtlinie haben, wählen Sie `Create an IAM-Rolle`, um eine zu erstellen.

13. Wenn Sie mit dem Ausfüllen des Formulars fertig sind, wählen Sie `Job erstellen`, um den Job zur Themenerkennung zu erstellen und zu starten.

Der neue Job wird in der Jobliste angezeigt, wobei das Statusfeld den Status des Jobs anzeigt. Das Feld kann `IN_PROGRESS` für einen Job verwendet werden, der gerade verarbeitet wird, `COMPLETED` für einen Job, der erfolgreich abgeschlossen wurde, und `FAILED` für einen Job, der einen Fehler aufweist. Sie können auf einen Job klicken, um weitere Informationen über den Job zu erhalten, einschließlich aller Fehlermeldungen.

Wenn der Job abgeschlossen ist, speichert Amazon Comprehend die Analyseergebnisse an dem Amazon S3-Ausgabespeicherort, den Sie für den Job angegeben haben. Eine Beschreibung der Analyseergebnisse finden Sie unter [Erkennung von PII-Entitäten](#)

## PII-Echtzeitanalyse (API)

Amazon Comprehend bietet synchrone API-Operationen in Echtzeit zur Analyse personenbezogener Daten (PII) in einem Dokument.

Themen

- [Lokalisieren von PII-Entitäten in Echtzeit \(API\)](#)
- [Kennzeichnung von PII-Echtzeit-Entitäten \(API\)](#)

## Lokalisieren von PII-Entitäten in Echtzeit (API)

Um personenbezogene Daten in einem einzelnen Dokument zu finden, können Sie den Amazon Comprehend [DetectPiiEntities](#) Comprehend-Vorgang verwenden. Ihr Eingabetext kann bis zu 100 Kilobyte an UTF-8-kodierten Zeichen enthalten. Zu den unterstützten Sprachen gehören Englisch und Spanisch.

### Auffinden von personenbezogenen Daten mithilfe von (CLI)

Das folgende Beispiel verwendet die `DetectPiiEntities` Operation mit dem AWS CLI.

Das Beispiel ist für Unix, Linux und macOS formatiert. Ersetzen Sie unter Windows den umgekehrten Schrägstrich (`\`), das Unix-Fortsetzungszeichen, am Ende jeder Zeile durch ein Caret-Zeichen oder Zirkumflex (`^`).

```
aws comprehend detect-pii-entities \  
  --text "Hello Paul Santos. The latest statement for your credit card \  
  account 1111-0000-1111-0000 was mailed to 123 Any Street, Seattle, WA \  
  98109." \  
  --language-code en
```

Amazon Comprehend antwortet wie folgt:

```
{  
  "Entities": [  
    {  
      "Score": 0.9999669790267944,  
      "Type": "NAME",  
      "BeginOffset": 6,  
      "EndOffset": 18  
    },  
    {  
      "Score": 0.8905550241470337,  
      "Type": "CREDIT_DEBIT_NUMBER",  
      "BeginOffset": 69,  
      "EndOffset": 88  
    },  
    {  
      "Score": 0.9999889731407166,  
      "Type": "ADDRESS",  
      "BeginOffset": 103,  
      "EndOffset": 138  
    }  
  ]  
}
```

```
    }
  ]
}
```

## Kennzeichnung von PII-Echtzeit-Entitäten (API)

Sie können synchrone API-Operationen in Echtzeit verwenden, um die Labels identifizierter PII-Entitätstypen zurückzugeben. Weitere Informationen finden Sie unter [Kennzeichnung von PII-Entitäten](#).

## Kennzeichnung von PII-Entitäten (CLI)

Das folgende Beispiel verwendet die `ContainsPiiEntities` Operation mit dem AWS CLI.

Das Beispiel ist für Unix, Linux und macOS formatiert. Ersetzen Sie unter Windows den umgekehrten Schrägstrich (\), das Unix-Fortsetzungszeichen, am Ende jeder Zeile durch ein Caret-Zeichen oder Zirkumflex (^).

```
aws comprehend contains-pii-entities \  
--text "Hello Paul Santos. The latest statement for your credit card \  
account 1111-0000-1111-0000 was mailed to 123 Any Street, Seattle, WA \  
98109." \  
--language-code en
```

Amazon Comprehend antwortet wie folgt:

```
{
  "Labels": [
    {
      "Name": "NAME",
      "Score": 0.9149109721183777
    },
    {
      "Name": "CREDIT_DEBIT_NUMBER",
      "Score": 0.8905550241470337
    }
  ]
}
```

# Asynchrone PII-Analyseaufträge (API)

## Asynchrone PII-Analyse (API)

Sie können asynchrone API-Operationen verwenden, um Analysejobs zu erstellen, um PII-Entitäten zu finden oder zu redigieren. Weitere Informationen zu PII-Entitätstypen finden Sie unter [Erkennung von PII-Entitäten](#)

### Themen

- [Suchen nach PII-Entitäten mit asynchronen Jobs \(API\)](#)
- [Redigieren von PII-Entitäten mit asynchronen Jobs \(API\)](#)

## Suchen nach PII-Entitäten mit asynchronen Jobs (API)

Führen Sie einen asynchronen Batch-Job aus, um personenbezogene Daten in einer Sammlung von Dokumenten zu finden. Um den Job auszuführen, laden Sie Ihre Dokumente auf Amazon S3 hoch und reichen Sie eine [StartPiiEntitiesDetectionJob](#)Anfrage ein.

### Themen

- [Bevor Sie beginnen](#)
- [Eingabeparameter](#)
- [Async-Job-Methoden](#)
- [Format der Ausgabedatei](#)
- [Asynchrone Analyse mit AWS Command Line Interface](#)

## Bevor Sie beginnen

Bevor Sie beginnen, stellen Sie sicher, dass Sie über Folgendes verfügen:

- Eingabe- und Ausgabe-Buckets — Identifizieren Sie die Amazon S3 S3-Buckets, die Sie für Eingabedateien und Ausgabedateien verwenden möchten. Die Buckets müssen sich in derselben Region befinden wie die API, die Sie aufrufen.
- IAM-Servicerolle — Sie benötigen eine IAM-Dienstrolle mit der Berechtigung, auf Ihre Eingabe- und Ausgabe-Buckets zuzugreifen. Weitere Informationen finden Sie unter [Rollenbasierte Berechtigungen, die für asynchrone Operationen erforderlich sind](#).



## Eingabeparameter

Geben Sie in Ihrer Anfrage die folgenden erforderlichen Parameter an:

- `InputDataConfig`— Geben Sie eine [InputDataConfig](#) Definition für Ihre Anfrage an, die die Eingabeeigenschaften für den Job enthält. Geben Sie für den `S3Uri` Parameter den Amazon S3 S3-Speicherort Ihrer Eingabedokumente an.
- `OutputDataConfig`— Geben Sie eine [OutputDataConfig](#) Definition für Ihre Anfrage an, die die Ausgabeeigenschaften für den Job enthält. Geben Sie für den `S3Uri` Parameter den Amazon S3 S3-Speicherort an, an den Amazon Comprehend die Ergebnisse seiner Analyse schreibt.
- `DataAccessRoleArn`— Geben Sie den Amazon-Ressourcennamen (ARN) einer AWS Identity and Access Management Rolle an. Diese Rolle muss Amazon Comprehend Lesezugriff auf Ihre Eingabedaten und Schreibzugriff auf Ihren Ausgabespeicherort in Amazon S3 gewähren. Weitere Informationen finden Sie unter [Rollenbasierte Berechtigungen, die für asynchrone Operationen erforderlich sind](#).
- `Mode`— Stellen Sie diesen Parameter auf ein. `ONLY_OFFSETS` Mit dieser Einstellung stellt die Ausgabe die Zeichen-Offsets bereit, mit denen jede PII-Entität im Eingabetext lokalisiert wird. Die Ausgabe umfasst auch Konfidenzwerte und PII-Entitätstypen.
- `LanguageCode`— Setzen Sie diesen Parameter auf `en` oder `es`. Amazon Comprehend unterstützt die Erkennung personenbezogener Daten in englischem oder spanischem Text.

## Async-Job-Methoden

Die `StartPiiEntitiesDetectionJob` gibt eine Job-ID zurück, sodass Sie den Fortschritt des Jobs überwachen und den Auftragsstatus abrufen können, wenn er abgeschlossen ist.

Um den Fortschritt eines Analyseauftrags zu überwachen, geben Sie die Job-ID für den [DescribePiiEntitiesDetectionJob](#) Vorgang an. Die Antwort von `DescribePiiEntitiesDetectionJob` enthält das `JobStatus` Feld mit dem aktuellen Status des Jobs. Ein erfolgreicher Job durchläuft die folgenden Zustände:

EINGEREICHT -> IN\_PROGRESS -> ABGESCHLOSSEN.

Verwenden `DescribePiiEntitiesDetectionJob` Sie diese Option, um den Speicherort der `JobStatus` Ergebnisse abzurufen, wenn ein Analysejob abgeschlossen, `FEHLGESCHLAGEN` oder `GESTOPPT` ist. Wenn der Auftragsstatus lautet `COMPLETED`, enthält die Antwort ein `OutputDataConfig` Feld, das ein Feld mit dem Amazon S3 S3-Speicherort der Ausgabedatei enthält.

Weitere Informationen zu den Schritten für die asynchrone Amazon Comprehend Analyse finden Sie unter. [Asynchrone Batchverarbeitung](#)

## Format der Ausgabedatei

Die Ausgabedatei verwendet den Namen der Eingabedatei, wobei am Ende „.out“ angehängt wird. Sie enthält die Ergebnisse der Analyse.

Im Folgenden finden Sie ein Beispiel für eine Ausgabedatei aus einem Analysejob, bei dem PII-Entitäten in Dokumenten erkannt wurden. Das Format der Eingabe ist ein Dokument pro Zeile.

```
{
  "Entities": [
    {
      "Type": "NAME",
      "BeginOffset": 40,
      "EndOffset": 69,
      "Score": 0.999995
    },
    {
      "Type": "ADDRESS",
      "BeginOffset": 247,
      "EndOffset": 253,
      "Score": 0.998828
    },
    {
      "Type": "BANK_ACCOUNT_NUMBER",
      "BeginOffset": 406,
      "EndOffset": 411,
      "Score": 0.693283
    }
  ],
  "File": "doc.txt",
  "Line": 0
},
{
  "Entities": [
    {
      "Type": "SSN",
      "BeginOffset": 1114,
      "EndOffset": 1124,
      "Score": 0.999999
    },
  ],
}
```

```
{
  "Type": "EMAIL",
  "BeginOffset": 3742,
  "EndOffset": 3775,
  "Score": 0.999993
},
{
  "Type": "PIN",
  "BeginOffset": 4098,
  "EndOffset": 4102,
  "Score": 0.999995
}
],
"File": "doc.txt",
"Line": 1
}
```

Im Folgenden finden Sie ein Beispiel für die Ausgabe einer Analyse, bei der das Eingabeformat ein Dokument pro Datei ist.

```
{
  "Entities": [
    {
      "Type": "NAME",
      "BeginOffset": 40,
      "EndOffset": 69,
      "Score": 0.999995
    },
    {
      "Type": "ADDRESS",
      "BeginOffset": 247,
      "EndOffset": 253,
      "Score": 0.998828
    },
    {
      "Type": "BANK_ROUTING",
      "BeginOffset": 279,
      "EndOffset": 289,
      "Score": 0.999999
    }
  ],
  "File": "doc.txt"
}
```

## Asynchrone Analyse mit AWS Command Line Interface

Das folgende Beispiel verwendet die `StartPiiEntitiesDetectionJob` Operation mit dem AWS CLI.

Das Beispiel ist für Unix, Linux und macOS formatiert. Ersetzen Sie unter Windows den umgekehrten Schrägstrich (`\`), das Unix-Fortsetzungszeichen, am Ende jeder Zeile durch ein Caret-Zeichen oder Zirkumflex (`^`).

```
aws comprehend start-pii-entities-detection-job \  
  --region region \  
  --job-name job name \  
  --cli-input-json file://path to JSON input file
```

Für den `cli-input-json` Parameter geben Sie den Pfad zu einer JSON-Datei an, die die Anforderungsdaten enthält, wie im folgenden Beispiel gezeigt.

```
{  
  "InputDataConfig": {  
    "S3Uri": "s3://input bucket/input path",  
    "InputFormat": "ONE_DOC_PER_LINE"  
  },  
  "OutputDataConfig": {  
    "S3Uri": "s3://output bucket/output path"  
  },  
  "DataAccessRoleArn": "arn:aws:iam::account ID:role/data access role"  
  "LanguageCode": "en",  
  "Mode": "ONLY_OFFSETS"  
}
```

Wenn die Anfrage zum Starten des Auftrags zur Erkennung von Ereignissen erfolgreich war, erhalten Sie eine Antwort, die der folgenden ähnelt:

```
{  
  "JobId": "5d2fbe6e...e2c"  
  "JobArn": "arn:aws:comprehend:us-west-2:123456789012:pii-entities-detection-  
job/5d2fbe6e...e2c"  
  "JobStatus": "SUBMITTED",  
}
```

Sie können den [DescribeEventsDetectionJob](#) Vorgang verwenden, um den Status eines vorhandenen Auftrags abzurufen. Wenn die Anfrage zum Starten des Auftrags zur Erkennung von Ereignissen erfolgreich war, erhalten Sie eine Antwort, die der folgenden ähnelt:

```
aws comprehend describe-pii-entities-detection-job \  
  --region region \  
  --job-id job ID
```

Wenn der Job erfolgreich abgeschlossen wurde, erhalten Sie eine Antwort, die der folgenden ähnelt:

```
{  
  "PiiEntitiesDetectionJobProperties": {  
    "JobId": "5d2fbe6e...e2c"  
    "JobArn": "arn:aws:comprehend:us-west-2:123456789012:pii-entities-detection-  
job/5d2fbe6e...e2c"  
    "JobName": "piiCLITest3",  
    "JobStatus": "COMPLETED",  
    "SubmitTime": "2022-05-05T14:54:06.169000-07:00",  
    "EndTime": "2022-05-05T15:00:17.007000-07:00",  
    "InputDataConfig": {  
      (identical to the input data that you provided with the request)  
    }  
  }  
}
```

## Redigieren von PII-Entitäten mit asynchronen Jobs (API)

Um die PII-Entitäten in Ihrem Text zu redigieren, starten Sie einen asynchronen Batch-Job. Um den Job auszuführen, laden Sie Ihre Dokumente auf Amazon S3 hoch und reichen Sie eine [StartPiiEntitiesDetectionJob](#) Anfrage ein.

### Themen

- [Bevor Sie beginnen](#)
- [Eingabeparameter](#)
- [Format der Ausgabedatei](#)
- [PII-Schwärzung mit dem AWS Command Line Interface](#)

## Bevor Sie beginnen

Bevor Sie beginnen, stellen Sie sicher, dass Sie über Folgendes verfügen:

- Eingabe- und Ausgabe-Buckets — Identifizieren Sie die Amazon S3 S3-Buckets, die Sie für Eingabedateien und Ausgabedateien verwenden möchten. Die Buckets müssen sich in derselben Region befinden wie die API, die Sie aufrufen.
- IAM-Servicerolle — Sie benötigen eine IAM-Dienstrolle mit der Berechtigung, auf Ihre Eingabe- und Ausgabe-Buckets zuzugreifen. Weitere Informationen finden Sie unter [Rollenbasierte Berechtigungen, die für asynchrone Operationen erforderlich sind](#).

## Eingabeparameter

Geben Sie in Ihrer Anfrage die folgenden erforderlichen Parameter an:

- `InputDataConfig`— Geben Sie eine [InputDataConfig](#) Definition für Ihre Anfrage an, die die Eingabeeigenschaften für den Job enthält. Geben Sie für den `S3Uri` Parameter den Amazon S3 S3-Speicherort Ihrer Eingabedokumente an.
- `OutputDataConfig`— Geben Sie eine [OutputDataConfig](#) Definition für Ihre Anfrage an, die die Ausgabeeigenschaften für den Job enthält. Geben Sie für den `S3Uri` Parameter den Amazon S3 S3-Speicherort an, an den Amazon Comprehend die Ergebnisse seiner Analyse schreibt.
- `DataAccessRoleArn`— Geben Sie den Amazon-Ressourcennamen (ARN) einer AWS Identity and Access Management Rolle an. Diese Rolle muss Amazon Comprehend Lesezugriff auf Ihre Eingabedaten und Schreibzugriff auf Ihren Ausgabespeicherort in Amazon S3 gewähren. Weitere Informationen finden Sie unter [Rollenbasierte Berechtigungen, die für asynchrone Operationen erforderlich sind](#).
- `Mode`— Stellen Sie diesen Parameter auf ein. `ONLY_REDACTION` Mit dieser Einstellung schreibt Amazon Comprehend eine Kopie Ihrer Eingabedokumente an den Ausgabespeicherort in Amazon S3. In dieser Kopie ist jede PII-Entität redigiert.
- `RedactionConfig`— Geben Sie eine [RedactionConfig](#) Definition für Ihre Anfrage an, die die Konfigurationsparameter für die Schwärzung enthält. Geben Sie die Arten von personenbezogenen Daten an, die geschwärzt werden sollen, und geben Sie an, ob jede PII-Entität durch den Namen ihres Typs oder ein Zeichen Ihrer Wahl ersetzt wird:
  - Geben Sie die PII-Entitätstypen an, die im Array geschwärzt werden sollen. `PiiEntityType` Um alle Entitätstypen zu schwärzen, setzen Sie den Array-Wert auf. `["ALL"]`
  - Um jede PII-Entität durch ihren Typ zu ersetzen, setzen Sie den `MaskMode` Parameter auf. `REPLACE_WITH_PII_ENTITY_TYPE` Mit dieser Einstellung wird beispielsweise die PII-Entität „Jane Doe“ durch „[NAME]“ ersetzt.

- Um die Zeichen in jeder PII-Entität durch ein Zeichen Ihrer Wahl zu ersetzen, setzen Sie den MaskMode Parameter auf MASK und setzen Sie den MaskCharacter Parameter auf das Ersatzzeichen. Geben Sie nur ein einziges Zeichen an. Gültige Zeichen sind! , #, \$, %, &, \* und @. Mit dieser Einstellung kann beispielsweise die PII-Entität „Jane Doe“ durch „\*\*\*\* \*“ ersetzt werden
- LanguageCode— Setzen Sie diesen Parameter auf en oder es. Amazon Comprehend unterstützt die Erkennung personenbezogener Daten in englischem oder spanischem Text.

## Format der Ausgabedatei

Das folgende Beispiel zeigt die Eingabe- und Ausgabedateien eines Analysejobs, bei dem personenbezogene Daten geschwärzt werden. Das Format der Eingabe ist ein Dokument pro Zeile.

```
{
Managing Your Accounts Primary Branch Canton John Doe Phone Number 443-573-4800 123
Main StreetBaltimore, MD 21224
Online Banking HowardBank.com Telephone 1-877-527-2703 Bank 3301 Boston Street,
Baltimore, MD 21224
```

Der Analyseauftrag zum Schwärzen dieser Eingabedatei erzeugt die folgende Ausgabedatei.

```
{
Managing Your Accounts Primary Branch ***** ***** Phone Number *****
*****
Online Banking ***** Telephone ***** Bank
*****
}
```

## PII-Schwärzung mit dem AWS Command Line Interface

Das folgende Beispiel verwendet die StartPiiEntitiesDetectionJob Operation mit dem. AWS CLI

Das Beispiel ist für Unix, Linux und macOS formatiert. Ersetzen Sie unter Windows den umgekehrten Schrägstrich (\), das Unix-Fortsetzungszeichen, am Ende jeder Zeile durch ein Caret-Zeichen oder Zirkumflex (^).

```
aws comprehend start-pii-entities-detection-job \
```

```
--region region \  
--job-name job name \  
--cli-input-json file://path to JSON input file
```

Für den `cli-input-json` Parameter geben Sie den Pfad zu einer JSON-Datei an, die die Anforderungsdaten enthält, wie im folgenden Beispiel gezeigt.

```
{  
  "InputDataConfig": {  
    "S3Uri": "s3://input bucket/input path",  
    "InputFormat": "ONE_DOC_PER_LINE"  
  },  
  "OutputDataConfig": {  
    "S3Uri": "s3://output bucket/output path"  
  },  
  "DataAccessRoleArn": "arn:aws:iam::account ID:role/data access role"  
  "LanguageCode": "en",  
  "Mode": "ONLY_REDACTION"  
  "RedactionConfig": {  
    "MaskCharacter": "*",  
    "MaskMode": "MASK",  
    "PiiEntityTypes": ["ALL"]  
  }  
}
```

Wenn die Anfrage zum Starten des Auftrags zur Erkennung von Ereignissen erfolgreich war, erhalten Sie eine Antwort, die der folgenden ähnelt:

```
{  
  "JobId": "7c4fbe6e...e5b"  
  "JobArn": "arn:aws:comprehend:us-west-2:123456789012:pii-entities-detection-  
job/7c4fbe6e...e5b"  
  "JobStatus": "SUBMITTED",  
}
```

Sie können den [DescribeEventsDetectionJob](#) Vorgang verwenden, um den Status eines vorhandenen Auftrags abzurufen.

```
aws comprehend describe-pii-entities-detection-job \  
--region region \  

```



```
--job-id job ID
```

Wenn der Job erfolgreich abgeschlossen wurde, erhalten Sie eine Antwort, die der folgenden ähnelt:

```
{
  "PiiEntitiesDetectionJobProperties": {
    "JobId": "7c4fbe6e...e5b"
    "JobArn": "arn:aws:comprehend:us-west-2:123456789012:pii-entities-detection-
job/7c4fbe6e...e5b"
    "JobName": "piiCLIredtest1",
    "JobStatus": "COMPLETED",
    "SubmitTime": "2022-05-05T14:54:06.169000-07:00",
    "EndTime": "2022-05-05T15:00:17.007000-07:00",
    "InputDataConfig": {
      (identical to the input data that you provided with the request)
    }
  }
}
```

# Dokumentenverarbeitung

Amazon Comprehend unterstützt die einstufige Dokumentenverarbeitung für die benutzerdefinierte Klassifizierung und die Erkennung benutzerdefinierter Entitäten. Sie können beispielsweise eine Mischung aus Klartextdokumenten und halbstrukturierten Dokumenten (wie PDF-Dokumenten, Microsoft Word-Dokumenten und Bildern) in einen benutzerdefinierten Analyseauftrag eingeben.

Bei Eingabedateien, die eine Textextraktion erfordern, führt Amazon Comprehend die Textextraktion automatisch durch, bevor die Analyse ausgeführt wird. Um den Textinhalt zu extrahieren, verwendet Amazon Comprehend einen internen Parser für native halbstrukturierte Dokumente und verwendet Amazon Textract APIs für Bilder und gescannte Dokumente.

Die Amazon Comprehend-Dokumentverarbeitung ist in jeder der Amazon Comprehend verfügbar [Unterstützte Regionen](#), außer Asien-Pazifik (Tokio) und AWS GovCloud (USA-West) unterstützt nur Klartextmodelle für die benutzerdefinierte Klassifizierung.

Die folgenden Themen enthalten Details zu den Eingabedokumenttypen, die Amazon Comprehend für benutzerdefinierte Analysen unterstützt.

## Themen

- [Eingaben für benutzerdefinierte Echtzeitanalysen](#)
- [Eingaben für asynchrone benutzerdefinierte Analysen](#)
- [Festlegen von Optionen für die Textextraktion](#)
- [Bewährte Methoden für Images](#)

## Eingaben für benutzerdefinierte Echtzeitanalysen

Bei der Echtzeitanalyse mit benutzerdefinierten Modellen wird ein einzelnes Dokument als Eingabe verwendet. In den folgenden Themen werden die Eingabedokumenttypen beschrieben, die Sie verwenden können.

## Themen

- [Klartextdokumente](#)
- [Semistrukturierte Dokumente](#)
- [Bilddateien und gescannte PDF-Dateien](#)
- [Amazon-Textract-Ausgabe](#)

- [Maximale Dokumentgrößen für Echtzeitanalysen](#)
- [Fehler in halbstrukturierten Dokumenten](#)

## Klartextdokumente

Geben Sie das Eingabedokument als Text im UTF-8-formatted.

## Semistrukturierte Dokumente

Zu den halbstrukturierten Dokumenten gehören native PDF-Dokumente und Word-Dokumente.

Standardmäßig verwendet die benutzerdefinierte Echtzeitanalyse den Amazon Comprehend-Parser, um den Text aus Word-Dateien und digitalen PDF-Dateien zu extrahieren. Für PDF-Dateien können Sie diesen Standard überschreiben und Amazon Textract verwenden, um den Text zu extrahieren. Siehe [Festlegen von Optionen für die Textextraktion](#).

## Bilddateien und gescannte PDF-Dateien

Zu den unterstützten Bildtypen gehören JPEG, PNG und TIFF.

Standardmäßig verwendet die benutzerdefinierte Entitätserkennung die Amazon TextractDetectDocumentText-API-Operation, um den Text aus Bilddateien und gescannten PDF-Dateien zu extrahieren. Sie können diesen Standard überschreiben, um stattdessen die AnalyzeDocument-API-Operation zu verwenden. Siehe [Festlegen von Optionen für die Textextraktion](#).

## Amazon-Textract-Ausgabe

Sie können die JSON-Ausgabe von der Amazon Textract DetectDocumentText API oder AnalyzeDocument API als Eingabe für die Echtzeit-API-Operationen für die benutzerdefinierte Klassifizierung und die Erkennung benutzerdefinierter Entitäten bereitstellen. Amazon Comprehend unterstützt diesen Eingabetyp für die Echtzeit-API-Operationen, aber nicht für die Konsole.

## Maximale Dokumentgrößen für Echtzeitanalysen

Für alle Eingabedokumenttypen beträgt das Maximum der Eingabedatei eine Seite mit nicht mehr als 10.000 Zeichen.

Die folgende Tabelle zeigt die maximalen Dateigrößen für Eingabedokumente.

Dateityp	Maximale Größe (API)	Maximale Größe (Konsole)
UTF-8-Textdokumente	10 KB	10 KB
PDF-Dokumente	10 MB	5 MB
Word-Dokumente	10 MB	1 MB
Abbildungsdateien	10 MB	5 MB
Textract-Ausgabedateien	1 MB	–

## Fehler in halbstrukturierten Dokumenten

Der - [ClassifyDocument](#) oder -[DetectEntities](#)API-Vorgang kann beim Extrahieren von Text aus einem halbstrukturierten Dokument oder einer Bilddatei auf Fehler auf Dokument- oder Seitenebene stoßen.

### Fehler auf Seitenebene

Wenn bei der - [ClassifyDocument](#) oder -[DetectEntities](#)API-Operation während der Verarbeitung einer Seite im Eingabedokument Fehler auftreten, enthält die API-Antwort für jeden Fehler einen Eintrag in der [Fehlerliste](#).

Der `ErrorCode` im Fehlerlisteneintrag enthält einen der folgenden Werte:

- `TEXTTRACT_BAD_PAGE` – Amazon Textract kann die Seite nicht lesen. Weitere Informationen zu Seitenlimits in Amazon Textract finden Sie unter [Seitenkontingente in Amazon Textract](#).
- `TEXTTRACT_PROVISIONED_THROUGHPUT_EXCEEDED` – Die Anzahl der Anfragen hat Ihr Durchsatzlimit überschritten. Weitere Informationen zu Durchsatzkontingenten in Amazon Textract finden Sie unter [Standardkontingente in Amazon Textract](#).
- `PAGE_CHARACTERS_EXCEEDED` – Zu viele Textzeichen auf der Seite (maximal 10.000 Zeichen).
- `PAGE_SIZE_EXCEEDED` – Die maximale Seitengröße beträgt 10 MB.
- `INTERNAL_SERVER_ERROR` – Bei der Anforderung ist ein Serviceproblem aufgetreten. Versuchen Sie die API-Anfrage erneut.

## Fehler auf Dokumentebene

Wenn die - [ClassifyDocument](#) oder - [DetectEntities](#) API-Operation einen Fehler auf Dokumentebene in Ihrem Eingabedokument erkennt, gibt die API eine `InvalidRequestException` Fehlerantwort zurück.

In der Fehlerantwort enthält das `Reason` Feld den Wert `INVALID_DOCUMENT`.

Das `Detail` Feld enthält einen der folgenden Werte:

- `DOCUMENT_SIZE_EXCEED` – Die Dokumentgröße ist zu groß. Überprüfen Sie die Größe Ihrer Datei und senden Sie die Anfrage erneut.
- `UNSUPPORTED_DOC_TYPE` – Der Dokumenttyp wird nicht unterstützt. Überprüfen Sie den Dateityp und senden Sie die Anforderung erneut.
- `PAGE_LIMIT_EXCEED` – Zu viele Seiten im Dokument. Überprüfen Sie die Anzahl der Seiten in Ihrer Datei und senden Sie die Anforderung erneut.
- `TEXTTRACT_ACCESS_DENIED_EXCEPTION` – Zugriff auf Amazon Textract verweigert. Stellen Sie sicher, dass Ihr Konto über die Berechtigung zur Verwendung der Amazon Textract- [DetectDocumentText](#) und [AnalyzeDocument](#)-API-Operationen verfügt, und übermitteln Sie die Anforderung erneut.

## Eingaben für asynchrone benutzerdefinierte Analysen

Sie können mehrere Dokumente in einen benutzerdefinierten asynchronen Analyseauftrag eingeben. In den folgenden Themen werden die Eingabedokumenttypen beschrieben, die Sie verwenden können. Die maximale Dateigröße variiert je nach Typ des Eingabedokuments.

Themen

- [Klartextdokumente](#)
- [Semistrukturierte Dokumente](#)
- [Bilddateien und gescannte PDF-Dateien](#)
- [Amazon-Textract-Ausgabe-JSON-Dateien](#)

## Klartextdokumente

Stellen Sie alle Klartext-Eingabedokumente als UTF-8-formatted Text bereit. In der folgenden Tabelle sind die maximalen Dateigrößen und andere Richtlinien aufgeführt.

### Note

Diese Grenzwerte gelten, wenn alle Eingabedateien Klartext sind.

Beschreibung	Kontingent/Anleitung
Maximale Dateigröße für ein Dokument pro Dateiformat (benutzerdefinierte Klassifizierung)	1 Byte–10 MB
Dokumentgröße (Kennung benutzerdefinierter Entitäten)	1 Byte–1 MB
Maximale Anzahl von Dateien, ein Dokument pro Datei	1 000 000
Maximale Anzahl von Zeilen, ein Dokument pro Zeile (für alle Dateien in Anforderung)	1 000 000
Dokumentkorpusgröße (alle Dokumente im Klartext kombiniert)	1 Byte–5 GB

## Semistrukturierte Dokumente

Zu den halbstrukturierten Dokumenten gehören native PDF-Dokumente und Word-Dokumente.

In der folgenden Tabelle sind die maximalen Dateigrößen und andere Richtlinien aufgeführt.

Beschreibung	Kontingent/Anleitung
Dokumentgröße (PDF)	1 Byte–50 MB
Dokumentgröße (Docx)	1 Byte–5 MB
Maximale Anzahl von Dateien	500

Beschreibung	Kontingent/Anleitung
Maximale Anzahl von Seiten für eine PDF- oder Docx-Datei	100
Dokumentkorporusgröße nach Textextraktion (Klartext, alle Dateien kombiniert)	1 Byte–5 GB

Standardmäßig verwendet die benutzerdefinierte Analyse den Amazon Comprehend-Parser, um den Text aus Word-Dateien und digitalen PDF-Dateien zu extrahieren. Für PDF-Dateien können Sie diesen Standard überschreiben und Amazon Textract verwenden, um den Text zu extrahieren. Siehe [Festlegen von Optionen für die Textextraktion](#).

## Bilddateien und gescannte PDF-Dateien

Benutzerdefinierte Analysen unterstützen JPEG-, PNG- und TIFF-Bilder.

In der folgenden Tabelle sind die maximalen Dateigrößen für Bilder aufgeführt. Gescannte PDF-Dateien unterliegen denselben maximalen Größen wie native PDF-Dateien.

Beschreibung	Kontingent/Anleitung
Bildgröße (JPG oder PNG)	1 Byte–10 MB
Bildgröße (TIFF)	1 Byte–10 MB. Maximal eine Seite.

Weitere Informationen zu Bildern finden Sie unter [Bewährte Methoden für Images](#).

Standardmäßig verwendet Amazon Comprehend die Amazon Textract DetectDocumentText API-Operation, um den Text aus Bilddateien und gescannten PDF-Dateien zu extrahieren. Sie können diesen Standard überschreiben, um stattdessen die AnalyzeDocument-API-Operation zu verwenden. Siehe [Festlegen von Optionen für die Textextraktion](#).

## Amazon-Textract-Ausgabe-JSON-Dateien

Für die Erkennung benutzerdefinierter Entitäten, aber nicht für die benutzerdefinierte Klassifizierung können Sie die Ausgabedatei aus der Amazon TextractAnalyzeDocument-API-Operation als Eingabe für Analyseaufträge bereitstellen.

## Festlegen von Optionen für die Textextraktion

Standardmäßig führt Amazon Comprehend die folgenden Aktionen aus, um Text aus einer Datei basierend auf dem Eingabedateityp zu extrahieren:

- Word-Dateien – Amazon Comprehend Parser extrahiert den Text.
- Digitale PDF-Dateien – Amazon Comprehend Parser extrahiert den Text.
- Bilddateien und gescannte PDF-Dateien – Amazon Comprehend verwendet die Amazon Textract DetectDocumentText API, um den Text zu extrahieren.

Für Bilddateien und PDF-Dateien können Sie den `DocumentReaderConfig` Parameter verwenden, um diese Standardextraktionsaktionen zu überschreiben. Dieser Parameter ist verfügbar, wenn Sie die Amazon Comprehend-Konsole oder -API für benutzerdefinierte Analysen in Echtzeit oder asynchron verwenden.

Der `DocumentReaderConfig` Parameter enthält drei Felder:

- `DocumentReadMode` – Legen Sie den Wert auf fest, damit `SERVICE_DEFAULT` Amazon Comprehend die Standardaktionen ausführen kann.

Legen Sie den Wert auf fest `FORCE_DOCUMENT_READ_ACTION`, um Amazon Textract zum Analysieren digitaler PDF-Dateien zu verwenden.

- `DocumentReadAction` – Legt die Amazon Textract API (`DetectDocumentText` oder `AnalyzeDocument`) fest, die verwendet werden soll, wenn Amazon Comprehend Amazon Textract für die Textextraktion verwendet.
- `FeatureTypes` – Wenn Sie `DocumentReadAction` für die Verwendung der `AnalyzeDocument` -API-Operation festlegen, können Sie eine oder beide der `FeatureTypes` (`TABLES`, `FORMS`) hinzufügen. Diese Funktionen enthalten zusätzliche Informationen zu den Tabellen und Formularen im -Dokument. Weitere Informationen zu diesen Funktionen finden Sie unter [Amazon Textract Document Analysis Response Objects](#).

Die folgenden Beispiele zeigen, wie für `DocumentReaderConfig` bestimmte Anwendungsfälle konfiguriert wird:

1. Verwenden Sie Amazon Textract für alle PDF-Dateien.
  - a. `DocumentReadMode` – Eingestellt auf `FORCE_DOCUMENT_READ_ACTION`.



- b. DocumentReadAction – Eingestellt auf `TEXTTRACT_DETECT_DOCUMENT_TEXT`.
  - c. FeatureTypes – Nicht erforderlich.
2. Verwenden Sie die Amazon Textract AnalyzeDocument API für alle PDF- und Bilddateien.
  - a. DocumentReadMode – Eingestellt auf `FORCE_DOCUMENT_READ_ACTION`.
  - b. DocumentReadAction – Eingestellt auf `TEXTTRACT_ANALYZE_DOCUMENT`.
  - c. FeatureTypes – Legen Sie auf `TABLES FORMS` oder beide Funktionen fest.
3. Verwenden Sie die Amazon Textract AnalyzeDocument API für gescannte PDF-Dateien und alle Bilddateien.
  - a. DocumentReadMode – Eingestellt auf `SERVICE_DEFAULT`.
  - b. DocumentReadAction – Eingestellt auf `TEXTTRACT_ANALYZE_DOCUMENT`.
  - c. FeatureTypes – Legen Sie auf `TABLES FORMS` oder beide Funktionen fest.

Weitere Informationen zu den Amazon-Textract-Optionen finden Sie unter [DocumentReaderConfig](#).

## Bewährte Methoden für Images

Wenn Sie Bilddateien für die benutzerdefinierte Klassifizierung oder die Erkennung benutzerdefinierter Entitäten verwenden, verwenden Sie die folgenden Richtlinien, um die besten Ergebnisse zu erzielen:

- Stellen Sie ein qualitativ hochwertiges Image bereit, idealerweise mindestens 150 DPI.
- Wenn die Bilddatei eines der unterstützten Formate (TIFF, JPEG oder PNG) verwendet, konvertieren oder reduzieren Sie die Datei nicht, bevor Sie sie in Amazon S3 hochladen.

Um beim Extrahieren von Text aus Tabellen in Dokumenten die besten Ergebnisse zu erzielen, folgen Sie diesen Methoden:

- Tabellen in Ihrem Dokument sind visuell von umgebenden Elementen auf der Seite getrennt. Beispielsweise wird die Tabelle nicht auf einem Bild oder komplexen Muster überlagert.
- Der Text in der Tabelle ist fein. Beispielsweise wird der Text nicht relativ zu anderem Text auf der Seite gedreht.

Beim Extrahieren von Text aus Tabellen können in den folgenden Fällen inkonsistente Ergebnisse angezeigt werden:

- Zusammengeführte Tabellenzellen umfassen mehrere Spalten.
- Tabellen haben Zellen, Zeilen oder Spalten, die sich von anderen Teilen derselben Tabelle unterscheiden.

# Benutzerdefinierte Klassifizierung

Verwenden Sie die benutzerdefinierte Klassifizierung, um Ihre Dokumente in Kategorien (Klassen) zu organisieren, die Sie definieren. Die benutzerdefinierte Klassifizierung ist ein zweistufiger Prozess. Zunächst trainieren Sie ein benutzerdefiniertes Klassifizierungsmodell (auch Klassifikator genannt), um die Klassen zu erkennen, die für Sie von Interesse sind. Anschließend verwenden Sie Ihr Modell, um eine beliebige Anzahl von Dokumentensätzen zu klassifizieren.

Sie können beispielsweise den Inhalt von Supportanfragen kategorisieren, sodass Sie die Anfrage an das richtige Support-Team weiterleiten können. Oder Sie können E-Mails, die Sie von Kunden erhalten haben, kategorisieren, um je nach Art der Kundenanfrage Hinweise zu geben. Sie können Amazon Comprehend mit Amazon Transcribe kombinieren, um Sprache in Text umzuwandeln und dann die Anfragen zu klassifizieren, die von Support-Telefonanrufen kommen.

Sie können die benutzerdefinierte Klassifizierung für ein einzelnes Dokument synchron (in Echtzeit) ausführen oder einen asynchronen Job starten, um eine Reihe von Dokumenten zu klassifizieren. Sie können mehrere benutzerdefinierte Klassifikatoren in Ihrem Konto haben, die jeweils mit unterschiedlichen Daten trainiert wurden. Die benutzerdefinierte Klassifizierung unterstützt eine Vielzahl von Eingabedokumenttypen, z. B. Klartext, PDF, Word und Bilder.

Wenn Sie einen Klassifizierungsauftrag einreichen, wählen Sie das zu verwendende Klassifizierungsmodell auf der Grundlage der Art der zu analysierenden Dokumente aus. Um beispielsweise Klartextdokumente zu analysieren, erzielen Sie die genauesten Ergebnisse, wenn Sie ein Modell verwenden, das Sie mit Klartextdokumenten trainiert haben. Um halbstrukturierte Dokumente (wie PDF, Word, Bilder, Amazon Textract Textract-Ausgabe oder gescannte Dateien) zu analysieren, erzielen Sie die genauesten Ergebnisse, wenn Sie ein Modell verwenden, das Sie mit systemeigenen Dokumenten trainiert haben.

## Themen

- [Vorbereitung der Trainingsdaten für Klassifikatoren](#)
- [Modelle zur Trainingsklassifizierung](#)
- [Analyse in Echtzeit ausführen](#)
- [Asynchrone Jobs ausführen](#)

# Vorbereitung der Trainingsdaten für Klassifikatoren

Für die benutzerdefinierte Klassifizierung trainieren Sie das Modell entweder im Mehrklassenmodus oder im Multi-Label-Modus. Der Mehrklassenmodus ordnet jedem Dokument eine einzelne Klasse zu. Der Modus mit mehreren Bezeichnungen ordnet jedem Dokument eine oder mehrere Klassen zu. Die Eingabedateiformate sind für jeden Modus unterschiedlich. Wählen Sie daher den zu verwendenden Modus aus, bevor Sie die Trainingsdaten erstellen.

## Note

Die Amazon Comprehend Comprehend-Konsole bezeichnet den Mehrklassenmodus als Single-Label-Modus.

Die benutzerdefinierte Klassifizierung unterstützt Modelle, die Sie mit Klartextdokumenten trainieren, und Modelle, die Sie mit systemeigenen Dokumenten (wie PDF, Word oder Bildern) trainieren. Weitere Informationen zu Klassifikatormodellen und den unterstützten Dokumenttypen finden Sie unter [Modelle zur Trainingsklassifizierung](#)

So bereiten Sie Daten für das Training eines benutzerdefinierten Klassifikatormodells vor:

1. Identifizieren Sie die Klassen, die dieser Klassifikator analysieren soll. Entscheiden Sie, welcher Modus verwendet werden soll (mehrere Klassen oder mehrere Beschriftungen).
2. Entscheiden Sie sich für den Modelltyp des Klassifikators, je nachdem, ob das Modell für die Analyse von Klartextdokumenten oder halbstrukturierten Dokumenten bestimmt ist.
3. Sammeln Sie Beispiele für Dokumente für jede der Klassen. Die Mindestanforderungen an die Schulung finden Sie unter [Allgemeine Kontingente für die Klassifizierung von Dokumenten](#).
4. Wählen Sie für ein Nur-Text-Modell das zu verwendende Trainingsdateiformat (CSV-Datei oder erweiterte Manifestdatei). Um ein systemeigenes Dokumentmodell zu trainieren, verwenden Sie immer eine CSV-Datei.

## Themen

- [Formate der Classifier-Trainingsdateien](#)
- [Mehrklassenmodus](#)
- [Modus mit mehreren Bezeichnungen](#)

## Formate der Classifier-Trainingsdateien

Für ein Klartextmodell können Sie Classifier-Trainingsdaten als CSV-Datei oder als erweiterte Manifestdatei bereitstellen, die Sie mit SageMaker Ground Truth erstellen. Die CSV-Datei oder erweiterte Manifestdatei enthält den Text für jedes Schulungsdokument und die zugehörigen Bezeichnungen.

Für ein systemeigenes Dokumentenmodell stellen Sie Classifier-Trainingsdaten als CSV-Datei bereit. Die CSV-Datei enthält den Dateinamen für jedes Schulungsdokument und die zugehörigen Labels. Sie fügen die Schulungsdokumente in den Amazon S3 S3-Eingabeordner für den Schulungsjob ein.

### CSV-Dateien

Sie stellen beschriftete Trainingsdaten als UTF-8-codierten Text in einer CSV-Datei bereit. Fügen Sie keine Kopfzeile hinzu. Das Hinzufügen einer Kopfzeile zu Ihrer Datei kann zu Laufzeitfehlern führen.

Für jede Zeile in der CSV-Datei enthält die erste Spalte eine oder mehrere Klassenbezeichnungen. Eine Klassenbezeichnung kann eine beliebige gültige UTF-8-Zeichenfolge sein. Wir empfehlen, klare Klassennamen zu verwenden, deren Bedeutung sich nicht überschneidet. Der Name kann Leerzeichen enthalten und aus mehreren Wörtern bestehen, die durch Unterstriche oder Bindestriche miteinander verbunden sind.

Lassen Sie vor oder nach den Kommas, die die Werte in einer Zeile trennen, keine Leerzeichen stehen.

Der genaue Inhalt der CSV-Datei hängt vom Klassifikatormodus und der Art der Trainingsdaten ab. Einzelheiten finden Sie in den Abschnitten zu [Mehrklassenmodus](#) und [Modus mit mehreren Bezeichnungen](#).

### Erweiterte Manifestdatei

Eine erweiterte Manifestdatei ist ein beschrifteter Datensatz, den Sie mit SageMaker Ground Truth erstellen. Ground Truth ist ein Datenkennzeichnungsdienst, der Ihnen — oder einer von Ihnen beschäftigten Belegschaft — dabei hilft, Trainingsdatensätze für Modelle des maschinellen Lernens zu erstellen.

Weitere Informationen zu Ground Truth und der damit generierten Ausgabe finden Sie unter [Use SageMaker Ground Truth to Label Data](#) im Amazon SageMaker Developer Guide.

Erweiterte Manifestdateien sind im JSON-Zeilenformat. In diesen Dateien ist jede Zeile ein vollständiges JSON-Objekt, das ein Schulungsdokument und die zugehörigen Beschriftungen enthält.

Der genaue Inhalt jeder Zeile hängt vom Klassifikatormodus ab. Einzelheiten finden Sie in den Abschnitten zu [Mehrklassenmodus](#) und [Modus mit mehreren Bezeichnungen](#).

Wenn Sie Amazon Comprehend Ihre Trainingsdaten zur Verfügung stellen, geben Sie einen oder mehrere Namen für Labelattribute an. Wie viele Attributnamen Sie angeben, hängt davon ab, ob Ihre erweiterte Manifestdatei das Ergebnis eines einzelnen Label-Jobs oder eines verketteten Labeling-Jobs ist.

Wenn Ihre Datei das Ergebnis eines einzelnen Label-Jobs ist, geben Sie den Namen des einzelnen Label-Attributs aus dem Ground Truth Job an.

Wenn Ihre Datei das Ergebnis eines verketteten Beschriftungsauftrags ist, geben Sie den Namen des Labelattributs für einen oder mehrere Jobs in der Kette an. Jeder Name eines Labelattributs enthält die Anmerkungen zu einem einzelnen Auftrag. Sie können bis zu 5 dieser Attribute für erweiterte Manifestdateien aus verketteten Labeling-Jobs angeben.

Weitere Informationen zu verketteten Labeling-Jobs und Beispiele für das Ergebnis, das sie produzieren, finden Sie unter [Chaining Labeling-Jobs](#) im Amazon SageMaker Developer Guide.

## Mehrklassenmodus

Im Mehrklassenmodus weist die Klassifizierung jedem Dokument eine Klasse zu. Die einzelnen Klassen schließen sich gegenseitig aus. Sie können beispielsweise einen Film als Komödie oder Science-Fiction klassifizieren, aber nicht als beides.

### Note

Die Amazon Comprehend Comprehend-Konsole bezeichnet den Mehrklassenmodus als Single-Label-Modus.

### Themen

- [Modelle im Klartext-Format](#)
- [Systemeigene Dokumentenmodelle](#)

## Modelle im Klartext-Format

Um ein Klartextmodell zu trainieren, können Sie beschriftete Trainingsdaten als CSV-Datei oder als erweiterte Manifestdatei von SageMaker Ground Truth bereitstellen.

## CSV-Datei

Allgemeine Informationen zur Verwendung von CSV-Dateien zum Trainieren von Klassifikatoren finden Sie unter [CSV-Dateien](#)

Stellen Sie die Trainingsdaten als zweispaltige CSV-Datei bereit. Für jede Zeile enthält die erste Spalte den Klassenbeschriftungswert. Die zweite Spalte enthält ein Beispiel-Textdokument für diese Klasse. Jede Zeile muss mit den Zeichen `\noder\ r\n`enden.

Das folgende Beispiel zeigt eine CSV-Datei mit drei Dokumenten.

```
CLASS,Text of document 1
CLASS,Text of document 2
CLASS,Text of document 3
```

Das folgende Beispiel zeigt eine Zeile einer CSV-Datei, die einen benutzerdefinierten Klassifikator dahingehend trainiert, zu erkennen, ob es sich bei einer E-Mail-Nachricht um Spam handelt:

```
SPAM,"Paulo, your $1000 award is waiting for you! Claim it while you still can at
http://example.com."
```

## Erweiterte Manifestdatei

Allgemeine Hinweise zur Verwendung erweiterter Manifestdateien für das Training von Klassifikatoren finden Sie unter [Erweiterte Manifestdatei](#).

Bei Klartextdokumenten ist jede Zeile der erweiterten Manifestdatei ein vollständiges JSON-Objekt, das ein Trainingsdokument, einen einzelnen Klassennamen und andere Metadaten von Ground Truth enthält. Das folgende Beispiel zeigt eine erweiterte Manifestdatei zum Trainieren eines benutzerdefinierten Klassifikators zur Erkennung von Spam-E-Mail-Nachrichten:

```
{"source":"Document 1 text", "MultiClassJob":0, "MultiClassJob-metadata":
{"confidence":0.62, "job-name":"labeling-job/multiclassjob", "class-name":"not_spam",
"human-annotated":"yes", "creation-date":"2020-05-21T17:36:45.814354",
"type":"groundtruth/text-classification"}}
{"source":"Document 2 text", "MultiClassJob":1, "MultiClassJob-metadata":
{"confidence":0.81, "job-name":"labeling-job/multiclassjob", "class-name":"spam",
"human-annotated":"yes", "creation-date":"2020-05-21T17:37:51.970530",
"type":"groundtruth/text-classification"}}
{"source":"Document 3 text", "MultiClassJob":1, "MultiClassJob-metadata":
{"confidence":0.81, "job-name":"labeling-job/multiclassjob", "class-name":"spam",
```

```
"human-annotated": "yes", "creation-date": "2020-05-21T17:37:51.970566",  
"type": "groundtruth/text-classification"}}
```

Das folgende Beispiel zeigt ein JSON-Objekt aus der erweiterten Manifestdatei, das aus Gründen der Lesbarkeit formatiert ist:

```
{  
  "source": "Paulo, your $1000 award is waiting for you! Claim it while you still can  
at http://example.com.",  
  "MultiClassJob": 0,  
  "MultiClassJob-metadata": {  
    "confidence": 0.98,  
    "job-name": "labeling-job/multiclassjob",  
    "class-name": "spam",  
    "human-annotated": "yes",  
    "creation-date": "2020-05-21T17:36:45.814354",  
    "type": "groundtruth/text-classification"  
  }  
}
```

In diesem Beispiel stellt das `source` Attribut den Text des Schulungsdokuments bereit, und das `MultiClassJob` Attribut weist den Index einer Klasse aus einer Klassifikationsliste zu. Das `job-name` Attribut ist der Name, den Sie für den Labeling-Job in Ground Truth definiert haben.

Wenn Sie den Classifier-Trainingsjob in Amazon Comprehend starten, geben Sie denselben Namen für den Labeling-Job an.

## Systemeigene Dokumentenmodelle

Ein systemeigenes Dokumentmodell ist ein Modell, das Sie mit systemeigenen Dokumenten (wie PDF, DOCX und Bildern) trainieren. Sie stellen die Trainingsdaten als CSV-Datei bereit.

### CSV-Datei

Allgemeine Informationen zur Verwendung von CSV-Dateien für Trainingsklassifikatoren finden Sie unter [CSV-Dateien](#).

Stellen Sie die Trainingsdaten als dreispaltige CSV-Datei bereit. Für jede Zeile enthält die erste Spalte den Klassenbeschriftungswert. Die zweite Spalte enthält den Dateinamen eines Beispieldokuments für diese Klasse. Die dritte Spalte enthält die Seitenzahl. Die Seitenzahl ist optional, wenn es sich bei dem Beispieldokument um ein Bild handelt.



Das folgende Beispiel zeigt eine CSV-Datei, die auf drei Eingabedokumente verweist.

```
CLASS,input-doc-1.pdf,3  
CLASS,input-doc-2.docx,1  
CLASS,input-doc-3.png
```

Das folgende Beispiel zeigt eine Zeile einer CSV-Datei, die einen benutzerdefinierten Klassifikator trainiert, um zu erkennen, ob es sich bei einer E-Mail-Nachricht um Spam handelt. Seite 2 der PDF-Datei enthält das Spam-Beispiel.

```
SPAM,email-content-3.pdf,2
```

## Modus mit mehreren Bezeichnungen

Im Modus mit mehreren Bezeichnungen stehen einzelne Klassen für unterschiedliche Kategorien, die sich nicht gegenseitig ausschließen. Bei der Klassifizierung mit mehreren Bezeichnungen werden jedem Dokument eine oder mehrere Klassen zugewiesen. Sie können beispielsweise einen Film als Dokumentarfilm und einen anderen als Science-Fiction, Action und Comedy klassifizieren.

Zu Schulungszwecken unterstützt der Multi-Label-Modus bis zu 1 Million Beispiele mit bis zu 100 eindeutigen Klassen.

### Themen

- [Modelle im Klartext-Format](#)
- [Systemeigene Dokumentenmodelle](#)

## Modelle im Klartext-Format

Um ein Klartextmodell zu trainieren, können Sie beschriftete Trainingsdaten als CSV-Datei oder als erweiterte Manifestdatei von SageMaker Ground Truth bereitstellen.

### CSV-Datei

Allgemeine Informationen zur Verwendung von CSV-Dateien zum Trainieren von Klassifikatoren finden Sie unter [CSV-Dateien](#)

Stellen Sie die Trainingsdaten als zweispaltige CSV-Datei bereit. Für jede Zeile enthält die erste Spalte die Klassenbeschriftungswerte und die zweite Spalte ein Beispieldokument für diese

Klassen. Um mehr als eine Klasse in die erste Spalte einzugeben, verwenden Sie ein Trennzeichen (z. B. ein |) zwischen den einzelnen Klassen.

```
CLASS,Text of document 1
CLASS,Text of document 2
CLASS|CLASS|CLASS,Text of document 3
```

Das folgende Beispiel zeigt eine Zeile einer CSV-Datei, die einen benutzerdefinierten Klassifikator trainiert, um Genres in Filmzusammenfassungen zu erkennen:

```
COMEDY|MYSTERY|SCIENCE_FICTION|TEEN,"A band of misfit teens become unlikely detectives when they discover troubling clues about their high school English teacher. Could the strange Mrs. Doe be an alien from outer space?"
```

Das Standardtrennzeichen zwischen Klassennamen ist ein senkrechter Strich (|). Sie können jedoch ein anderes Zeichen als Trennzeichen verwenden. Das Trennzeichen muss sich von allen Zeichen in Ihren Klassennamen unterscheiden. Wenn Ihre Klassen beispielsweise CLASS\_1, CLASS\_2 und CLASS\_3 sind, ist der Unterstrich (\_) Teil des Klassennamens. Verwenden Sie also keinen Unterstrich als Trennzeichen für die Trennung von Klassennamen.

## Erweiterte Manifestdatei

Allgemeine Hinweise zur Verwendung erweiterter Manifestdateien für das Training von Klassifikatoren finden Sie unter [Erweiterte Manifestdatei](#).

Bei Klartextdokumenten ist jede Zeile der erweiterten Manifestdatei ein vollständiges JSON-Objekt. Es enthält ein Schulungsdokument, Klassennamen und andere Metadaten von Ground Truth. Das folgende Beispiel ist eine erweiterte Manifestdatei, mit der ein benutzerdefinierter Klassifikator trainiert wird, um Genres in Filmzusammenfassungen zu erkennen:

```
{"source":"Document 1 text", "MultiLabelJob":[0,4], "MultiLabelJob-metadata":{"job-name":"labeling-job/multilabeljob", "class-map":{"0":"action", "4":"drama"}, "human-annotated":"yes", "creation-date":"2020-05-21T19:02:21.521882", "confidence-map":{"0":0.66}, "type":"groundtruth/text-classification-multilabel"}}
{"source":"Document 2 text", "MultiLabelJob":[3,6], "MultiLabelJob-metadata":{"job-name":"labeling-job/multilabeljob", "class-map":{"3":"comedy", "6":"horror"}, "human-annotated":"yes", "creation-date":"2020-05-21T19:00:01.291202", "confidence-map":{"1":0.61,"0":0.61}, "type":"groundtruth/text-classification-multilabel"}}
{"source":"Document 3 text", "MultiLabelJob":[1], "MultiLabelJob-metadata":{"job-name":"labeling-job/multilabeljob", "class-map":{"1":"action"}, "human-
```

```
annotated": "yes", "creation-date": "2020-05-21T18:58:51.662050", "confidence-map": {"1": 0.68}, "type": "groundtruth/text-classification-multilabel"}}
```

Das folgende Beispiel zeigt ein JSON-Objekt aus der erweiterten Manifestdatei, das aus Gründen der Lesbarkeit formatiert ist:

```
{
  "source": "A band of misfit teens become unlikely detectives when
            they discover troubling clues about their high school English
teacher.
            Could the strange Mrs. Doe be an alien from outer space?",
  "MultiLabelJob": [
    3,
    8,
    10,
    11
  ],
  "MultiLabelJob-metadata": {
    "job-name": "labeling-job/multilabeljob",
    "class-map": {
      "3": "comedy",
      "8": "mystery",
      "10": "science_fiction",
      "11": "teen"
    },
    "human-annotated": "yes",
    "creation-date": "2020-05-21T19:00:01.291202",
    "confidence-map": {
      "3": 0.95,
      "8": 0.77,
      "10": 0.83,
      "11": 0.92
    },
    "type": "groundtruth/text-classification-multilabel"
  }
}
```

In diesem Beispiel stellt das `source` Attribut den Text des Schulungsdokuments bereit, und das `MultiLabelJob` Attribut weist die Indizes mehrerer Klassen aus einer Klassifikationsliste zu. Der Jobname in den `MultiLabelJob` Metadaten ist der Name, den Sie für den Labeling-Job in Ground Truth definiert haben.

## Systemeigene Dokumentenmodelle

Ein systemeigenes Dokumentmodell ist ein Modell, das Sie mit systemeigenen Dokumenten (wie PDF-, DOCX- und Bilddateien) trainieren. Sie stellen beschriftete Trainingsdaten als CSV-Datei bereit.

### CSV-Datei

Allgemeine Informationen zur Verwendung von CSV-Dateien für Trainingsklassifikatoren finden Sie unter [CSV-Dateien](#).

Stellen Sie die Trainingsdaten als dreispaltige CSV-Datei bereit. Für jede Zeile enthält die erste Spalte die Klassenbeschriftungswerte. Die zweite Spalte enthält den Dateinamen eines Beispieldokuments für diese Klassen. Die dritte Spalte enthält die Seitenzahl. Die Seitenzahl ist optional, wenn es sich bei dem Beispieldokument um ein Bild handelt.

Um mehr als eine Klasse in die erste Spalte einzugeben, verwenden Sie ein Trennzeichen (z. B. ein |) zwischen den einzelnen Klassen.

```
CLASS,input-doc-1.pdf,3  
CLASS,input-doc-2.docx,1  
CLASS|CLASS|CLASS,input-doc-3.png,2
```

Das folgende Beispiel zeigt eine Zeile einer CSV-Datei, die einen benutzerdefinierten Klassifikator trainiert, um Genres in Filmzusammenfassungen zu erkennen. Seite 2 der PDF-Datei enthält das Beispiel einer Komödie oder eines Jugendfilms.

```
COMEDY|TEEN,movie-summary-1.pdf,2
```

Das Standardtrennzeichen zwischen Klassennamen ist ein senkrechter Strich (|). Sie können jedoch ein anderes Zeichen als Trennzeichen verwenden. Das Trennzeichen muss sich von allen Zeichen in Ihren Klassennamen unterscheiden. Wenn Ihre Klassen beispielsweise CLASS\_1, CLASS\_2 und CLASS\_3 sind, ist der Unterstrich (\_) Teil des Klassennamens. Verwenden Sie also keinen Unterstrich als Trennzeichen für die Trennung von Klassennamen.

## Modelle zur Trainingsklassifizierung

Um ein Modell für die benutzerdefinierte Klassifizierung zu trainieren, definieren Sie die Kategorien und stellen Beispieldokumente bereit, um das benutzerdefinierte Modell zu trainieren. Sie trainieren

das Modell entweder im Modus mit mehreren Klassen oder im Modus mit mehreren Bezeichnungen. Im Mehrklassenmodus wird jedem Dokument eine einzelne Klasse zugeordnet. Der Modus mit mehreren Bezeichnungen ordnet jedem Dokument eine oder mehrere Klassen zu.

Die benutzerdefinierte Klassifizierung unterstützt zwei Arten von Klassifizierungsmodellen: Klartextmodelle und native Dokumentenmodelle. Ein Nur-Text-Modell klassifiziert Dokumente anhand ihres Textinhalts. Ein systemeigenes Dokumentenmodell klassifiziert Dokumente auch anhand des Textinhalts. Ein systemeigenes Dokumentenmodell kann auch zusätzliche Signale verwenden, z. B. aus dem Layout des Dokuments. Sie trainieren ein systemeigenes Dokumentmodell mit systemeigenen Dokumenten, damit das Modell die Layoutinformationen lernt.

Klartextmodelle weisen die folgenden Eigenschaften auf:

- Sie trainieren das Modell mit UTF-8-codierten Textdokumenten.
- Sie können das Modell anhand von Dokumenten in einer der folgenden Sprachen trainieren: Englisch, Spanisch, Deutsch, Italienisch, Französisch oder Portugiesisch.
- Die Schulungsdokumente für einen bestimmten Klassifikator müssen alle dieselbe Sprache verwenden.
- Die Schulungsdokumente sind Klartext, sodass für die Textextraktion keine zusätzlichen Gebühren anfallen.

Systemeigene Dokumentenmodelle weisen die folgenden Eigenschaften auf:

- Sie trainieren das Modell mithilfe halbstrukturierter Dokumente, zu denen die folgenden Dokumenttypen gehören:
  - Digitale und gescannte PDF-Dokumente.
  - Word-Dokumente (DOCX).
  - Bilder: JPG-Dateien, PNG-Dateien und einseitige TIFF-Dateien.
  - Textract-API-Ausgabe von JSON-Dateien.
- Sie trainieren das Modell anhand englischer Dokumente.
- Wenn Ihre Schulungsunterlagen gescannte Dokumentdateien enthalten, fallen zusätzliche Gebühren für die Textextraktion an. Weitere Informationen finden Sie auf der Seite [Amazon Comprehend Pricing](#).

Sie können jeden der unterstützten Dokumenttypen anhand eines der beiden Modelltypen klassifizieren. Für genaueste Ergebnisse empfehlen wir jedoch, ein Klartextmodell zur

Klassifizierung von Klartextdokumenten und ein systemeigenes Dokumentmodell zur Klassifizierung halbstrukturierter Dokumente zu verwenden.

## Themen

- [Trainieren Sie benutzerdefinierte Klassifikatoren \(Konsole\)](#)
- [Trainieren Sie benutzerdefinierte Klassifikatoren \(API\)](#)
- [Testen Sie die Trainingsdaten](#)
- [Klassifizieren Sie die Trainingsleistung](#)
- [Benutzerdefinierte Klassifikator-Metriken](#)

## Trainieren Sie benutzerdefinierte Klassifikatoren (Konsole)

Sie können mithilfe der Konsole einen benutzerdefinierten Klassifikator erstellen und trainieren und dann den benutzerdefinierten Klassifikator verwenden, um Ihre Dokumente zu analysieren.


Um einen benutzerdefinierten Klassifikator zu trainieren, benötigen Sie eine Reihe von Schulungsdokumenten. Sie kennzeichnen diese Dokumente mit den Kategorien, die der Dokumentenklassifizierer erkennen soll. Informationen zur Vorbereitung Ihrer Schulungsdokumente finden Sie unter [Vorbereitung der Trainingsdaten für Klassifikatoren](#).

So erstellen und trainieren Sie ein Klassifizierungsmodell für Dokumente

1. [Melden Sie sich bei der Amazon Comprehend Comprehend-Konsole an AWS Management Console und öffnen Sie sie unter <https://console.aws.amazon.com/comprehend/>](#)
2. Wählen Sie im linken Menü Anpassung und dann Benutzerdefinierte Klassifizierung aus.
3. Wählen Sie Neues Modell erstellen.
4. Geben Sie unter Modelleinstellungen einen Modellnamen für den Klassifikator ein. Der Name muss innerhalb Ihres Kontos und Ihrer aktuellen Region eindeutig sein.  
  
(Optional) Geben Sie einen Versionsnamen ein. Der Name muss innerhalb Ihres Kontos und Ihrer aktuellen Region eindeutig sein.
5. Wählen Sie die Sprache der Schulungsunterlagen aus. Informationen zu den Sprachen, die Classifier unterstützen, finden Sie unter [Modelle zur Trainingsklassifizierung](#).
6. (Optional) Wenn Sie die Daten auf dem Speichervolume verschlüsseln möchten, während Amazon Comprehend Ihren Trainingsjob verarbeitet, wählen Sie Classifier-Verschlüsselung.

Wählen Sie dann aus, ob Sie einen KMS-Schlüssel verwenden möchten, der mit Ihrem aktuellen Konto verknüpft ist, oder einen von einem anderen Konto.

- Wenn Sie einen Schlüssel verwenden, der dem aktuellen Konto zugeordnet ist, wählen Sie die Schlüssel-ID für die KMS-Schlüssel-ID.
- Wenn Sie einen Schlüssel verwenden, der einem anderen Konto zugeordnet ist, geben Sie den ARN für die Schlüssel-ID unter KMS-Schlüssel-ARN ein.

 Note

Weitere Informationen zur Erstellung und Verwendung von KMS-Schlüsseln und der zugehörigen Verschlüsselung finden Sie unter [AWS Key Management Service \(AWS KMS\)](#).

7. Wählen Sie unter Datenspezifikationen den zu verwendenden Trainingsmodelltyp aus.
  - Nur-Text-Dokumente: Wählen Sie diese Option, um ein Nur-Text-Modell zu erstellen. Trainieren Sie das Modell mithilfe von Klartextdokumenten.
  - Systemeigene Dokumente: Wählen Sie diese Option, um ein systemeigenes Dokumentmodell zu erstellen. Trainieren Sie das Modell mit systemeigenen Dokumenten (PDF, Word, Bilder).
8. Wählen Sie das Datenformat Ihrer Trainingsdaten. Informationen zu den Datenformaten finden Sie unter [Formate der Classifier-Trainingsdateien](#).
  - CSV-Datei: Wählen Sie diese Option, wenn Ihre Trainingsdaten das CSV-Dateiformat verwenden.
  - Erweitertes Manifest: Wählen Sie diese Option, wenn Sie Ground Truth verwendet haben, um erweiterte Manifestdateien für Ihre Trainingsdaten zu erstellen. Dieses Format ist verfügbar, wenn Sie Nur-Text-Dokumente als Trainingsmodelltyp ausgewählt haben.
9. Wählen Sie den zu verwendenden Klassifizierermodus aus.
  - Einzellabelmodus: Wählen Sie diesen Modus, wenn sich die Kategorien, die Sie Dokumenten zuweisen, gegenseitig ausschließen und Sie Ihren Klassifizierer darauf trainieren, jedem Dokument ein Etikett zuzuweisen. In der Amazon Comprehend API wird der Single-Label-Modus als Mehrklassenmodus bezeichnet.

- Multi-Label-Modus: Wählen Sie diesen Modus, wenn mehrere Kategorien gleichzeitig auf ein Dokument angewendet werden können und Sie Ihren Klassifizierer darauf trainieren, jedem Dokument ein oder mehrere Labels zuzuweisen.
10. Wenn Sie den Modus „Mehrere Beschriftungen“ wählen, können Sie das Trennzeichen für Beschriftungen auswählen. Verwenden Sie dieses Trennzeichen, um Bezeichnungen voneinander zu trennen, wenn es mehrere Klassen für ein Schulungsdokument gibt. Das Standardtrennzeichen ist der senkrechte Strich.
  11. (Optional) Wenn Sie Augmented Manifest als Datenformat ausgewählt haben, können Sie bis zu fünf Augmented Manifest-Dateien eingeben. Jede erweiterte Manifest-Datei enthält entweder einen Trainingsdatensatz oder einen Testdatensatz. Sie müssen mindestens einen Trainingsdatensatz angeben. Testdatensätze sind optional. Gehen Sie wie folgt vor, um die erweiterten Manifestdateien zu konfigurieren:
    - a. Erweitern Sie unter Trainings- und Testdatensatz den Bereich Eingabeposition.
    - b. Wählen Sie unter Datensatztyp die Option Trainingsdaten oder Testdaten aus.
    - c. Geben Sie für den S3-Speicherort der erweiterten SageMaker Ground Truth Manifest-Datei den Speicherort des Amazon S3 S3-Buckets ein, der die Manifestdatei enthält, oder navigieren Sie zu diesem, indem Sie Browse S3 wählen. Die IAM-Rolle, die Sie für die Zugriffsberechtigungen für den Trainingsjob verwenden, muss über Leseberechtigungen für den S3-Bucket verfügen.
    - d. Geben Sie für die Attributnamen den Namen des Attributs ein, das Ihre Anmerkungen enthält. Wenn die Datei Anmerkungen aus mehreren verketteten Beschriftungsaufträgen enthält, fügen Sie für jeden Auftrag ein Attribut hinzu.
    - e. Um einen weiteren Eingabeort hinzuzufügen, wählen Sie Eingabeposition hinzufügen und konfigurieren Sie dann den nächsten Speicherort.
  12. (Optional) Wenn Sie die CSV-Datei als Datenformat ausgewählt haben, gehen Sie wie folgt vor, um den Trainingsdatensatz und den optionalen Testdatensatz zu konfigurieren:
    - a. Geben Sie unter Trainingsdatensatz den Speicherort des Amazon S3 S3-Buckets ein, der Ihre CSV-Datei mit Trainingsdaten enthält, oder navigieren Sie zu diesem, indem Sie Browse S3 wählen. Die IAM-Rolle, die Sie für die Zugriffsberechtigungen für den Trainingsjob verwenden, muss über Leseberechtigungen für den S3-Bucket verfügen.

(Optional) Wenn Sie Native Dokumente als Trainingsmodelltyp wählen, geben Sie auch die URL des Amazon S3 S3-Ordners an, der die Trainingsbeispieldateien enthält.



b. Wählen Sie unter Testdatensatz aus, ob Sie zusätzliche Daten für Amazon Comprehend bereitstellen, um das trainierte Modell zu testen.

- Autosplit: Autosplit wählt automatisch 10% Ihrer Trainingsdaten aus, um sie für die Verwendung als Testdaten zu reservieren.
- (Optional) Vom Kunden bereitgestellt: Geben Sie die URL der Testdaten-CSV-Datei in Amazon S3 ein. Sie können auch zu seinem Speicherort in Amazon S3 navigieren und Ordner auswählen wählen.

(Optional) Wenn Sie Native Dokumente als Trainingsmodelltyp wählen, geben Sie auch die URL des Amazon S3 S3-Ordners an, der die Testdateien enthält.

13. (Optional) Im Lesemodus für Dokumente können Sie die standardmäßigen Textextraktionsaktionen überschreiben. Diese Option ist für Klartext-Modelle nicht erforderlich, da sie für die Textextraktion für gescannte Dokumente gilt. Weitere Informationen finden Sie unter [Festlegen von Optionen für die Textextraktion](#).

14. (Optional für Klartextmodelle) Geben Sie unter Ausgabedaten den Speicherort eines Amazon S3 S3-Buckets ein, um Trainingsausgabedaten wie die Konfusionsmatrix zu speichern. Weitere Informationen finden Sie unter [Verwechslungsmatrix](#).

(Optional) Wenn Sie das Ausgabeergebnis Ihres Trainingsjobs verschlüsseln möchten, wählen Sie Verschlüsselung. Wählen Sie dann aus, ob Sie einen KMS-Schlüssel verwenden möchten, der mit dem aktuellen Konto verknüpft ist, oder einen von einem anderen Konto.

- Wenn Sie einen Schlüssel verwenden, der dem aktuellen Konto zugeordnet ist, wählen Sie den Schlüsselalias für die KMS-Schlüssel-ID.
- Wenn Sie einen Schlüssel verwenden, der einem anderen Konto zugeordnet ist, geben Sie den ARN für den Schlüsselalias oder die Schlüssel-ID unter KMS-Schlüssel-ID ein.

15. Wählen Sie für die IAM-Rolle die Option Eine bestehende IAM-Rolle auswählen und wählen Sie dann eine bestehende IAM-Rolle aus, die über Leseberechtigungen für den S3-Bucket verfügt, der Ihre Schulungsdokumente enthält. Die Rolle muss über eine Vertrauensrichtlinie verfügen, die mit 1 beginnt, um gültig `comprehend.amazonaws.com` zu sein.

Wenn Sie noch keine IAM-Rolle mit diesen Berechtigungen haben, wählen Sie Create an IAM-Rolle aus, um eine zu erstellen. Wählen Sie die Zugriffsberechtigungen aus, die Sie dieser Rolle gewähren möchten, und wählen Sie dann ein Namenssuffix, um die Rolle von den IAM-Rollen in Ihrem Konto zu unterscheiden.

**Note**

Für verschlüsselte Eingabedokumente muss die verwendete IAM-Rolle ebenfalls über eine entsprechende Berechtigung verfügen. `kms:Decrypt` Weitere Informationen finden Sie unter [Erforderliche Berechtigungen für die Verwendung der KMS-Verschlüsselung](#).

16. (Optional) Um Ihre Ressourcen von einer VPC aus in Amazon Comprehend zu starten, geben Sie die VPC-ID unter VPC ein oder wählen Sie die ID aus der Dropdownliste aus.
  1. Wählen Sie das Subnetz unter Subnetze (en) aus. Nachdem Sie das erste Subnetz ausgewählt haben, können Sie weitere auswählen.
  2. Wählen Sie unter Sicherheitsgruppe (n) die zu verwendende Sicherheitsgruppe aus, falls Sie eine angegeben haben. Nachdem Sie die erste Sicherheitsgruppe ausgewählt haben, können Sie weitere auswählen.

**Note**

Wenn Sie eine VPC mit Ihrem Klassifizierungsjob verwenden, müssen die für die Vorgänge Create und Start DataAccessRole verwendeten VPC über Berechtigungen für die VPC verfügen, die auf die Eingabedokumente und den Ausgabe-Bucket zugreift.

17. (Optional) Um dem benutzerdefinierten Klassifikator ein Tag hinzuzufügen, geben Sie unter Tags ein Schlüssel-Wert-Paar ein. Wählen Sie Add tag. Um dieses Paar vor der Erstellung des Klassifikators zu entfernen, wählen Sie Tag entfernen aus. Weitere Informationen finden Sie unter [Markieren Ihrer -Ressourcen](#).
18. Wählen Sie Erstellen.

In der Konsole wird die Seite „Klassifikatoren“ angezeigt. Der neue Klassifikator wird in der Tabelle Submitted als Status angezeigt. Wenn der Klassifikator mit der Verarbeitung der Schulungsdokumente beginnt, ändert sich der Status in Training Wenn ein Klassifikator einsatzbereit ist, ändert sich der Status in oder. Trained Trained with warnings Wenn der Status lautet TRAINED\_WITH\_WARNINGS, überprüfen Sie den Ordner mit übersprungenen Dateien im. [Klassifizieren Sie die Trainingsleistung](#)

Wenn Amazon Comprehend bei der Erstellung oder Schulung auf Fehler gestoßen ist, ändert sich der Status auf. In `error` Sie können einen Klassifikator-Job in der Tabelle auswählen, um weitere Informationen über den Klassifikator, einschließlich aller Fehlermeldungen, zu erhalten.

Classifiers				
<input type="text" value="Search"/> <span>Stop</span> <span>Copy</span> <span>Delete</span> <span>Manage tags</span> <span>Create job</span> <span>Train classifier</span>				
				All ▾
				< 1 > ⚙️
Name ▾	Training started ▾	Training ended ▾	Status ▾	
<input type="radio"/> <a href="#">classifiertags5-copy</a>	7/22/2019, 3:48:38 PM	7/22/2019, 3:57:18 PM	✔️ Trained	
<input type="radio"/> <a href="#">classifiertags5</a>	6/24/2019, 3:40:28 PM	6/24/2019, 3:47:26 PM	✔️ Trained	
<input type="radio"/> <a href="#">classifiertags!</a>	6/3/2019, 6:33:16 PM	6/3/2019, 6:33:35 PM	❌ In error	
<input type="radio"/> <a href="#">hk-classifier-output-2</a>	4/9/2019, 11:28:26 AM	4/9/2019, 11:28:29 AM	❌ In error	

## Trainieren Sie benutzerdefinierte Klassifikatoren (API)

Verwenden Sie die [CreateDocumentClassifier](#) Operation, um einen benutzerdefinierten Klassifikator zu erstellen und zu trainieren.

Sie können den Fortschritt der Anfrage mithilfe der [DescribeDocumentClassifier](#) Operation überwachen. Nachdem das Status Feld zu gewechselt ist `TRAINED`, können Sie den Klassifikator verwenden, um Dokumente zu klassifizieren. Wenn der Status lautet `TRAINED_WITH_WARNINGS`, überprüfen Sie den Ordner mit den übersprungenen Dateien im Ordner [Klassifizieren Sie die Trainingsleistung](#) aus dem Vorgang. `CreateDocumentClassifier`

### Themen

- [Schulung der benutzerdefinierten Klassifizierung mit dem AWS Command Line Interface](#)
- [Verwenden des AWS SDK for Java oder SDK für Python](#)

## Schulung der benutzerdefinierten Klassifizierung mit dem AWS Command Line Interface

Die folgenden Beispiele zeigen, wie Sie den `CreateDocumentClassifier` Vorgang, den `DescribeDocumentClassificationJob` Vorgang und andere benutzerdefinierte Klassifikator-APIs mit dem AWS CLI verwenden.

Die Beispiele sind für Unix, Linux und macOS formatiert. Ersetzen Sie unter Windows den umgekehrten Schrägstrich (\), das Unix-Fortsetzungszeichen, am Ende jeder Zeile durch ein Caret-Zeichen oder Zirkumflex (^).

Erstellen Sie mithilfe der Operation einen benutzerdefinierten Klartext-Klassifikator. `create-document-classifier`

```
aws comprehend create-document-classifier \  
  --region region \  
  --document-classifier-name testDelete \  
  --language-code en \  
  --input-data-config S3Uri=s3://S3Bucket/docclass/file name \  
  --data-access-role-arn arn:aws:iam::account number:role/testFlywheelDataAccess
```

Um einen systemeigenen benutzerdefinierten Klassifikator zu erstellen, geben Sie in der Anfrage die folgenden zusätzlichen Parameter an. `create-document-classifier`

1. `DocumentType`: Setzen Sie den Wert auf `SEMI_STRUCTURED_DOCUMENT`.
2. `Documents`: Der S3-Speicherort für die Schulungsdokumente (und optional die Testdokumente).
3. `OutputDataConfig`: Geben Sie den S3-Speicherort für die Ausgabedokumente (und einen optionalen KMS-Schlüssel) an.
4. `DocumentReaderConfig`: Optionales Feld für Einstellungen zur Textextraktion.

```
aws comprehend create-document-classifier \  
  --region region \  
  --document-classifier-name testDelete \  
  --language-code en \  
  --input-data-config  
    S3Uri=s3://S3Bucket/docclass/file name \  
    DocumentType \  
    Documents \  
  --output-data-config S3Uri=s3://S3Bucket/docclass/file name \  
  --data-access-role-arn arn:aws:iam::account number:role/testFlywheelDataAccess
```

Rufen Sie mithilfe der Operation Informationen zu einem benutzerdefinierten Klassifikator mit dem Dokumentenklassifikator ARN ab. `DescribeDocumentClassifier`

```
aws comprehend describe-document-classifier \  

```

```
--region region \  
--document-classifier-arn arn:aws:comprehend:region:account number:document-  
classifier/file name
```

Löschen Sie mithilfe der Operation einen benutzerdefinierten Klassifikator.

### DeleteDocumentClassifier

```
aws comprehend delete-document-classifier \  
--region region \  
--document-classifier-arn arn:aws:comprehend:region:account number:document-  
classifier/testDelete
```

Listet alle benutzerdefinierten Klassifikatoren in dem Konto auf, das den ListDocumentClassifiers Vorgang verwendet.

```
aws comprehend list-document-classifiers  
--region region
```

## Verwenden des AWS SDK for Java oder SDK für Python

SDK-Beispiele zum Erstellen und Trainieren eines benutzerdefinierten Klassifikators finden Sie unter [Verwendung CreateDocumentClassifier mit einem AWS SDK oder CLI](#).

## Testen Sie die Trainingsdaten

Nach dem Training des Modells testet Amazon Comprehend das benutzerdefinierte Klassifikatormodell. Wenn Sie keinen Testdatensatz bereitstellen, trainiert Amazon Comprehend das Modell mit 90 Prozent der Trainingsdaten. Es reserviert 10 Prozent der Trainingsdaten für Tests. Wenn Sie einen Testdatensatz bereitstellen, müssen die Testdaten mindestens ein Beispiel für jedes eindeutige Label im Trainingsdatensatz enthalten.

Durch das Testen des Modells erhalten Sie Metriken, anhand derer Sie die Genauigkeit des Modells abschätzen können. Die Konsole zeigt die Metriken im Abschnitt Classifier-Performance der Classifier-Detailseite in der Konsole an. Sie werden auch in den Metrics Feldern zurückgegeben, die durch den [DescribeDocumentClassifier](#) Vorgang zurückgegeben wurden.

Im folgenden Beispiel für Trainingsdaten gibt es fünf Bezeichnungen: DOCUMENTARY, DOCUMENTARY, SCIENCE\_FICTION, DOCUMENTARY, ROMANTIC\_COMEDY. Es gibt drei einzigartige Klassen: DOCUMENTARY, SCIENCE\_FICTION, ROMANTIC\_COMEDY.

Spalte 1	Spalte 2
DOKUMENTARFILM	Text des Dokuments 1
DOKUMENTARFILM	Text des Dokuments 2
SCIENCE_FICTION	Text des Dokuments 3
DOKUMENTARFILM	Text des Dokuments 4
ROMANTISCHE_KOMÖDIE	Text des Dokuments 5

Bei der auto Aufteilung (bei der Amazon Comprehend 10 Prozent der Trainingsdaten für Tests reserviert) kann der Testdatensatz keine Beispiele für dieses Label enthalten, wenn die Trainingsdaten begrenzte Beispiele für ein bestimmtes Label enthalten. Wenn der Trainingsdatensatz beispielsweise 1000 Instanzen der DOCUMENTAR-Klasse, 900 Instanzen von SCIENCE\_FICTION und eine einzelne Instanz der Klasse ROMANTIC\_COMEDY enthält, kann der Testdatensatz 100 DOCUMENTARY- und 90 SCIENCE\_FICTION-Instanzen enthalten, aber keine ROMANTIC\_COMEDY-Instanzen, da nur ein einziges Beispiel verfügbar ist.

Nachdem Sie das Training Ihres Modells abgeschlossen haben, liefern die Trainingsmetriken Informationen, anhand derer Sie entscheiden können, ob das Modell für Ihre Anforderungen ausreichend genau ist.

## Klassifizieren Sie die Trainingsleistung

Nachdem Amazon Comprehend das Training des benutzerdefinierten Klassifikatormodells abgeschlossen hat, erstellt es Ausgabedateien im Amazon S3-Ausgabespeicherort, den Sie in der [CreateDocumentClassifier](#) API-Anforderung oder der entsprechenden Konsolenanforderung angegeben haben.

Amazon Comprehend erstellt eine Konfusionsmatrix, wenn Sie ein Klartext-Modell oder ein systemeigenes Dokumentenmodell trainieren. Es kann zusätzliche Ausgabedateien erstellen, wenn Sie ein systemeigenes Dokumentenmodell trainieren.

### Themen

- [Verwechslungsmatrix](#)
- [Zusätzliche Ausgaben für native Dokumentenmodelle](#)

## Verwechslungsmatrix

Wenn Sie ein benutzerdefiniertes Klassifikatormodell trainieren, erstellt Amazon Comprehend eine Konfusionsmatrix, die Kennzahlen darüber enthält, wie gut das Modell beim Training abgeschnitten hat. Diese Matrix zeigt eine Matrix von Beschriftungen, die das Modell vorhergesagt hat, im Vergleich zu den tatsächlichen Dokumentenbeschriftungen. Amazon Comprehend verwendet einen Teil der Trainingsdaten, um die Konfusionsmatrix zu erstellen.

Eine Konfusionsmatrix gibt Aufschluss darüber, welche Klassen mehr Daten verwenden könnten, um die Modelleistung zu verbessern. Eine Klasse mit einem hohen Anteil an korrekten Vorhersagen hat die höchste Anzahl von Ergebnissen entlang der Diagonalen der Matrix. Wenn die Zahl auf der Diagonalen eine niedrigere Zahl ist, hat die Klasse einen geringeren Anteil richtiger Vorhersagen. Sie können weitere Trainingsbeispiele für diese Klasse hinzufügen und das Modell erneut trainieren. Wenn beispielsweise 40 Prozent der Stichproben mit Label A als Label D klassifiziert werden, verbessert das Hinzufügen weiterer Proben für Label A und Label D die Leistung des Klassifikators.

Nachdem Amazon Comprehend das Klassifikatormodell erstellt hat, ist die Konfusionsmatrix in der `confusion_matrix.json` Datei im S3-Ausgabespeicherort verfügbar.

Das Format der Konfusionsmatrix variiert, je nachdem, ob Sie Ihren Klassifikator im Mehrklassenmodus oder im Multi-Label-Modus trainiert haben.

### Themen

- [Konfusionsmatrix für den Mehrklassenmodus](#)
- [Konfusionsmatrix für den Modus mit mehreren Labels](#)

### Konfusionsmatrix für den Mehrklassenmodus

Im Mehrklassenmodus schließen sich die einzelnen Klassen gegenseitig aus, sodass bei der Klassifizierung jedem Dokument eine Bezeichnung zugewiesen wird. Ein Tier kann beispielsweise ein Hund oder eine Katze sein, aber nicht beides gleichzeitig.

Betrachten Sie das folgende Beispiel für eine Konfusionsmatrix für einen trainierten Klassifikator mit mehreren Klassen:

```
A B X Y <-(predicted label)
A 1 2 0 4
B 0 3 0 1
X 0 0 1 0
Y 1 1 1 1
```

```
^
|
(actual label)
```

In diesem Fall prognostizierte das Modell Folgendes:

- Ein „A“ -Label wurde genau vorhergesagt, zwei „A“ -Labels wurden fälschlicherweise als „B“ -Labels vorhergesagt und vier „A“ -Labels wurden fälschlicherweise als „Y“ -Label vorhergesagt.
- Drei „B“ -Labels wurden genau vorhergesagt, und ein „B“ -Label wurde fälschlicherweise als „Y“ -Label vorhergesagt.
- Ein „X“ wurde genau vorhergesagt.
- Ein „Y“ -Label wurde genau vorhergesagt, eines wurde fälschlicherweise als „A“ -Label vorhergesagt, eines wurde fälschlicherweise als „B“ -Label vorhergesagt und eines wurde fälschlicherweise als „X“ -Label vorhergesagt.

Die diagonale Linie in der Matrix (A:A, B:B, X:X und Y:Y) zeigt die genauen Vorhersagen. Die Vorhersagefehler sind die Werte außerhalb der Diagonale. In diesem Fall zeigt die Matrix die folgenden Vorhersagefehlerraten:

- A-Etiketten: 86%
- B-Etiketten: 25%
- X-Etiketten: 0%
- Y-Etiketten: 75%

Der Klassifikator gibt die Konfusionsmatrix als Datei im JSON-Format zurück. Die folgende JSON-Datei stellt die Matrix für das vorherige Beispiel dar.

```
{
  "type": "multi_class",
  "confusion_matrix": [
    [1, 2, 0, 4],
    [0, 3, 0, 1],
    [0, 0, 1, 0],
    [1, 1, 1, 1]],
  "labels": ["A", "B", "X", "Y"],
  "all_labels": ["A", "B", "X", "Y"]
}
```



## Konfusionsmatrix für den Modus mit mehreren Labels

Im Modus mit mehreren Bezeichnungen kann die Klassifizierung einem Dokument eine oder mehrere Klassen zuweisen. Betrachten Sie das folgende Beispiel einer Konfusionsmatrix für einen aus mehreren Klassen bestehenden, trainierten Klassifikator.

In diesem Beispiel gibt es drei mögliche Bezeichnungen: ComedyAction, und. Drama Die Konfusionsmatrix mit mehreren Bezeichnungen erstellt für jedes Etikett eine 2x2-Matrix.

Comedy	Action		Drama		<-(predicted label)
	No	Yes	No	Yes	
No	2	1	No	1	1
Yes	0	2	Yes	2	1
^    ------(was this label actually used)-----					

In diesem Fall gab das Modell für das Etikett Folgendes zurück: Comedy

- Zwei Fälle, in denen das Vorhandensein eines Comedy Labels genau vorhergesagt wurde. Wirklich positiv (TP).
- Zwei Fälle, in denen genau vorhergesagt wurde, dass ein Comedy Etikett fehlt. Richtig negativ (TN).
- Keine Fälle, in denen fälschlicherweise vorhergesagt wurde, dass ein Comedy Etikett vorhanden ist. Falsch positiv (FP).
- Ein Fall, in dem fälschlicherweise vorhergesagt wurde, dass ein Comedy Etikett nicht vorhanden ist. Falsch negativ (FN).

Wie bei einer Konfusionsmatrix mit mehreren Klassen zeigt die diagonale Linie in jeder Matrix die genauen Vorhersagen.

In diesem Fall hat das Modell Comedy Labels in 80% der Fälle (TP plus TN) genau und in 20% der Fälle falsch vorhergesagt (FP plus FN).

Der Klassifikator gibt die Konfusionsmatrix als Datei im JSON-Format zurück. Die folgende JSON-Datei stellt die Matrix für das vorherige Beispiel dar.

```
{
  "type": "multi_label",
  "confusion_matrix": [
    [[2, 1],
     [0, 2]],
    [[1, 1],
     [2, 1]],
    [[3, 0],
     [1, 1]]
  ],
  "labels": ["Comedy", "Action", "Drama"]
  "all_labels": ["Comedy", "Action", "Drama"]
}
```

## Zusätzliche Ausgaben für native Dokumentenmodelle

Amazon Comprehend kann zusätzliche Ausgabedateien erstellen, wenn Sie ein systemeigenes Dokumentenmodell trainieren.

### Amazon Textract Textract-Ausgabe

Wenn Amazon Comprehend die Amazon Textract Textract-APIs aufgerufen hat, um Text für eines der Schulungsdokumente zu extrahieren, speichert es die Amazon Textract Textract-Ausgabedateien im S3-Ausgabespeicherort. Es verwendet die folgende Verzeichnisstruktur:

- Schulungsunterlagen:

```
amazon-textract-output/train/<file_name>/<page_num>/textract_output.json
```

- Testdokumente:

```
amazon-textract-output/test/<file_name>/<page_num>/textract_output.json
```

Amazon Comprehend füllt den Testordner aus, wenn Sie Testdokumente in der API-Anfrage angegeben haben.

### Fehler bei der Anmerkung zu Dokumenten

Amazon Comprehend erstellt die folgenden Dateien im Amazon S3 S3-Ausgabespeicherort (im Ordner `skipped_documents/`), falls Anmerkungen fehlschlagen:

- `failed_annotations_train.jsonl`

Die Datei ist vorhanden, falls Anmerkungen in den Trainingsdaten fehlgeschlagen sind.

- `failed_annotations_test.jsonl`

Die Datei ist vorhanden, wenn die Anfrage Testdaten enthielt und alle Anmerkungen in den Testdaten fehlgeschlagen sind.

Die fehlgeschlagenen Annotationsdateien sind JSONL-Dateien mit dem folgenden Format:

```
{
  "File": "String", "Page": Number, "ErrorCode": "...", "ErrorMessage": "..."}
{"File": "String", "Page": Number, "ErrorCode": "...", "ErrorMessage": "..."}
}
```

## Benutzerdefinierte Klassifikator-Metriken

Amazon Comprehend bietet Kennzahlen, mit denen Sie abschätzen können, wie gut ein benutzerdefinierter Klassifikator abschneidet. Amazon Comprehend berechnet die Metriken anhand der Testdaten aus dem Classifier-Trainingsjob. Die Metriken stellen die Leistung des Modells während des Trainings genau dar, sodass sie ungefähr der Modellleistung für die Klassifizierung ähnlicher Daten entsprechen.

Verwenden Sie API-Operationen, [DescribeDocumentClassifier](#) um z. B. die Metriken für einen benutzerdefinierten Klassifikator abzurufen.

### Note

Weitere Informationen zu den zugrunde liegenden [Precision-, Recall- und F1-Score-Metriken finden Sie unter Metriken](#): Precision, Recall und FScore. Diese Kennzahlen werden auf Klassenebene definiert. Amazon Comprehend verwendet Makro-Mittelwertbildung, um diese Metriken zu den Testsätzen P, R und F1 zu kombinieren, wie im Folgenden beschrieben.

### Themen

- [Metriken](#)
- [Verbesserung der Leistung Ihres benutzerdefinierten Klassifikators](#)

## Metriken

Amazon Comprehend unterstützt die folgenden Metriken:

### Themen

- [Accuracy](#)
- [Präzision \(Makro-Präzision\)](#)
- [Rückruf \(Makrorückruf\)](#)
- [F1-Score \(Makro-F1-Score\)](#)
- [Hamming-Verlust](#)
- [Mikro-Präzision](#)
- [Mikro-Recall](#)
- [Mikro-F1-Score](#)

Um die Metriken für einen Classifier anzuzeigen, öffnen Sie die Seite Classifier Details in der Konsole.

Classifier performance <a href="#">Info</a>			
Accuracy	Precision	Recall	F1 score
0.34	0.3298	0.3304	0.32
Hamming loss	Micro precision	Micro recall	Micro F1 score
-	-	-	-

### Accuracy

Die Genauigkeit gibt den Prozentsatz der Labels aus den Testdaten an, die das Modell genau vorhergesagt hat. Um die Genauigkeit zu berechnen, dividieren Sie die Anzahl der genau vorhergesagten Etiketten in den Testdokumenten durch die Gesamtzahl der Etiketten in den Testdokumenten.

### Beispiel

Tatsächliches Etikett	Vorhergesagtes Label	Genau/Falsch
1	1	Genau
0	1	Falsch
2	3	Falsch
3	3	Genau
2	2	Genau
1	1	Genau
3	3	Genau

Die Genauigkeit besteht aus der Anzahl der genauen Vorhersagen geteilt durch die Anzahl der gesamten Teststichproben =  $5/7 = 0,714$  oder 71,4%

### Präzision (Makro-Präzision)

Präzision ist ein Maß für die Nützlichkeit der Klassifikatorergebnisse in den Testdaten. Sie ist definiert als die Anzahl der korrekt klassifizierten Dokumente geteilt durch die Gesamtzahl der Klassifizierungen für die Klasse. Hohe Genauigkeit bedeutet, dass der Klassifikator deutlich relevantere Ergebnisse als irrelevante Ergebnisse lieferte.

Die Precision Metrik wird auch als Makrogenauigkeit bezeichnet.

Das folgende Beispiel zeigt Präzisionsergebnisse für einen Testsatz.

Label (Bezeichnung)	Größe der Stichprobe	Präzision des Etiketts
Label_1	400	0.75
Bezeichnung_2	300	0,80
Bezeichnung_3	30000	0.90
Bezeichnung_4	20	0.50

Label (Bezeichnung)	Größe der Stichprobe	Präzision des Etiketts
Bezeichnung_5	10	0,40

Die Messgröße Precision (Macro Precision) für das Modell lautet daher:

$$\text{Macro Precision} = (0.75 + 0.80 + 0.90 + 0.50 + 0.40)/5 = 0.67$$

### Rückruf (Makrorückruf)

Dies gibt den Prozentsatz der richtigen Kategorien in Ihrem Text an, den das Modell vorhersagen kann. Diese Kennzahl ergibt sich aus der Mittelung der Erinnerungswerte aller verfügbaren Labels. Der Rückruf ist ein Maß dafür, wie vollständig die Klassifizierungsergebnisse für die Testdaten sind.

Ein hoher Erinnerungswert bedeutet, dass der Klassifikator die meisten relevanten Ergebnisse zurückgegeben hat.

Die Recall Metrik wird auch als Macro Recall bezeichnet.

Das folgende Beispiel zeigt die Rückrufergebnisse für einen Testsatz.

Label (Bezeichnung)	Größe der Stichprobe	Rückruf des Etiketts
Label_1	400	0,70
Etikett_2	300	0,70
Bezeichnung_3	30000	0,98
Bezeichnung_4	20	0,80
Bezeichnung_5	10	0.10

Die Recall-Metrik (Macro Recall) für das Modell lautet daher:

$$\text{Macro Recall} = (0.70 + 0.70 + 0.98 + 0.80 + 0.10)/5 = 0.656$$

## F1-Score (Makro-F1-Score)

Der F1-Score wird aus den Recall Werten Precision und abgeleitet. Es misst die Gesamtgenauigkeit des Klassifikators. Die höchste Punktzahl ist 1 und die niedrigste Punktzahl ist 0.

Amazon Comprehend berechnet den Macro F1 Score. Es ist der ungewichtete Durchschnitt der Label-F1-Scores. Verwenden Sie den folgenden Testsatz als Beispiel:

Label (Bezeichnung)	Größe der Stichprobe	Etikett: F1-Score
Bezeichnung_1	400	0,724
Etikett_2	300	0,824
Aufkleber_3	30000	0,94
Bezeichnung_4	20	0,62
Bezeichnung_5	10	0,16

Der F1-Score (Macro F1 Score) für das Modell wird wie folgt berechnet:

$$\text{Macro F1 Score} = (0.724 + 0.824 + 0.94 + 0.62 + 0.16)/5 = 0.6536$$

## Hamming-Verlust

Der Anteil der Labels, die falsch vorhergesagt wurden. Wird auch als Anteil falscher Labels im Vergleich zur Gesamtzahl der Labels angesehen. Werte, die näher an Null liegen, sind besser.

## Mikro-Präzision

Original:

Ähnlich der Präzisionsmetrik, mit der Ausnahme, dass die Mikrogenauigkeit auf der Gesamtpunktzahl aller Präzisionswerte zusammengenommen basiert.

## Mikro-Recall

Ähnlich wie bei der Recall-Metrik, mit der Ausnahme, dass der Mikro-Recall auf der Gesamtpunktzahl aller zusammengefassten Erinnerungswerte basiert.

## Mikro-F1-Score

Der Micro F1-Score ist eine Kombination aus den Kennzahlen Micro Precision und Micro Recall.

## Verbesserung der Leistung Ihres benutzerdefinierten Klassifikators

Die Metriken geben Aufschluss darüber, wie Ihr benutzerdefinierter Klassifikator während eines Klassifizierungsjobs abschneidet. Wenn die Metriken niedrig sind, ist das Klassifizierungsmodell für Ihren Anwendungsfall möglicherweise nicht effektiv. Sie haben mehrere Möglichkeiten, die Leistung Ihres Klassifikators zu verbessern:

1. Geben Sie in Ihren Trainingsdaten konkrete Beispiele an, die eine klare Trennung der Kategorien definieren. Stellen Sie beispielsweise Dokumente bereit, die eindeutige Wörter/Sätze verwenden, um die Kategorie darzustellen.
2. Füge weitere Daten für unterrepräsentierte Labels in deinen Trainingsdaten hinzu.
3. Versuche, die Verzerrung in den Kategorien zu reduzieren. Wenn das größte Etikett in Ihren Daten mehr als das Zehnfache der Dokumente im kleinsten Etikett enthält, versuchen Sie, die Anzahl der Dokumente für das kleinste Etikett zu erhöhen. Achten Sie darauf, das Schrägverhältnis zwischen den am stärksten repräsentierten und den am wenigsten repräsentierten Klassen auf höchstens 10:1 zu reduzieren. Sie können auch versuchen, Eingabedokumente aus den am stärksten vertretenen Klassen zu entfernen.

## Analyse in Echtzeit ausführen

Nachdem Sie einen benutzerdefinierten Klassifikator trainiert haben, können Sie Dokumente mithilfe von Echtzeitanalysen klassifizieren. Bei der Echtzeitanalyse wird ein einzelnes Dokument als Eingabe verwendet und die Ergebnisse werden synchron zurückgegeben. Die benutzerdefinierte Klassifizierung akzeptiert eine Vielzahl von Dokumenttypen als Eingaben für die Echtzeitanalyse. Details hierzu finden Sie unter [Eingaben für benutzerdefinierte Echtzeitanalysen](#).

Wenn Sie Bilddateien oder gescannte PDF-Dokumente analysieren möchten, muss Ihre IAM-Richtlinie die Erlaubnis zur Verwendung von zwei Amazon Textract Textract-API-Methoden (DetectDocumentText und AnalyzeDocument) gewähren. Amazon Comprehend ruft diese Methoden während der Textextraktion auf. Eine Beispielrichtlinie finden Sie unter [Erforderliche Berechtigungen zum Ausführen von Dokumentanalyseaktionen](#).

Sie müssen einen Endpunkt erstellen, um Echtzeitanalysen mithilfe eines benutzerdefinierten Klassifizierungsmodells durchzuführen.



## Themen

- [Echtzeitanalyse für die benutzerdefinierte Klassifizierung \(Konsole\)](#)
- [Echtzeitanalyse für benutzerdefinierte Klassifizierung \(API\)](#)
- [Ergebnisse für Echtzeitanalysen](#)

## Echtzeitanalyse für die benutzerdefinierte Klassifizierung (Konsole)

Sie können die Amazon Comprehend verwenden, um Echtzeitanalysen mithilfe eines benutzerdefinierten Klassifizierungsmodells durchzuführen.

Sie erstellen einen Endpunkt, um die Echtzeitanalyse auszuführen. Ein Endpunkt umfasst verwaltete Ressourcen, die Ihr benutzerdefiniertes Modell für Inferenzen in Echtzeit verfügbar machen.

Informationen zur Bereitstellung des Endpunktdurchsatzes und zu den damit verbundenen Kosten finden Sie unter [Verwenden von Amazon Comprehend-Endpunkten](#)

## Themen

- [Einen Endpunkt für die benutzerdefinierte Klassifizierung erstellen](#)
- [Ausführung einer benutzerdefinierten Klassifizierung in Echtzeit](#)

## Einen Endpunkt für die benutzerdefinierte Klassifizierung erstellen

Um einen Endpunkt (Konsole) zu erstellen

1. [Melden Sie sich bei der Amazon Comprehend Comprehend-Konsole an AWS Management Console und öffnen Sie sie unter <https://console.aws.amazon.com/comprehend/>](#)
2. Wählen Sie im linken Menü Endpoints und klicken Sie auf die Schaltfläche Create Endpoint. Ein Bildschirm „Endpoint erstellen“ wird geöffnet.
3. Geben Sie dem Endpunkt einen Namen. Der Name muss innerhalb der aktuellen Region und des aktuellen Kontos eindeutig sein.
4. Wählen Sie ein benutzerdefiniertes Modell aus, an das Sie den neuen Endpunkt anhängen möchten. In der Dropdownliste können Sie nach dem Modellnamen suchen.

**Note**

Sie müssen ein Modell erstellen, bevor Sie ihm einen Endpunkt zuordnen können. Wenn Sie noch kein Modell haben, finden Sie weitere Informationen unter [Modelle zur Trainingsklassifizierung](#).

5. (Optional) Um dem Endpunkt ein Tag hinzuzufügen, geben Sie unter Tags ein Schlüssel-Wert-Paar ein und wählen Sie Tag hinzufügen aus. Um dieses Paar vor der Erstellung des Endpunkts zu entfernen, wählen Sie Tag entfernen
6. Geben Sie die Anzahl der Inferenzeinheiten (IU) ein, die dem Endpunkt zugewiesen werden sollen. Jede Einheit entspricht einem Durchsatz von 100 Zeichen pro Sekunde für bis zu zwei Dokumente pro Sekunde. Informationen zum Endpunktdurchsatz finden Sie unter [Verwenden von Amazon Comprehend-Endpunkten](#).
7. (Optional) Wenn Sie einen neuen Endpunkt erstellen, haben Sie die Möglichkeit, den IU-Schätzer zu verwenden. Abhängig vom Durchsatz oder der Anzahl der Zeichen, die Sie pro Sekunde analysieren möchten, kann es schwierig sein, zu wissen, wie viele Inferenzeinheiten Sie benötigen. Dieser optionale Schritt kann Ihnen dabei helfen, die Anzahl der anzufordernden Einheiten zu bestimmen.
8. Prüfen Sie in der Kaufübersicht Ihre geschätzten stündlichen, täglichen und monatlichen Endpunktkosten.
9. Aktivieren Sie das Kontrollkästchen, wenn Sie wissen, dass Ihr Konto vom Start bis zum Löschen Gebühren für den Endpunkt anfallen.
10. Wählen Sie Endpunkt erstellen

## Ausführung einer benutzerdefinierten Klassifizierung in Echtzeit

Sobald Sie einen Endpunkt erstellt haben, können Sie mithilfe Ihres benutzerdefinierten Modells eine Echtzeitanalyse durchführen. Es gibt zwei Möglichkeiten, Echtzeitanalysen von der Konsole aus auszuführen. Sie können Text eingeben oder eine Datei hochladen, wie im Folgenden gezeigt.

Um eine Echtzeitanalyse mit einem benutzerdefinierten Modell (Konsole) durchzuführen

1. [Melden Sie sich bei der Amazon Comprehend Comprehend-Konsole an AWS Management Console und öffnen Sie sie unter https://console.aws.amazon.com/comprehend/](#)
2. Wählen Sie im linken Menü die Option Echtzeitanalyse aus.

3. Wählen Sie unter Eingabetyp die Option Benutzerdefiniert als Analysetyp aus.
4. Wählen Sie unter Benutzerdefinierter Modelltyp die Option Benutzerdefinierte Klassifizierung aus.
5. Wählen Sie unter Endpunkt den Endpunkt aus, den Sie verwenden möchten. Dieser Endpunkt ist mit einem bestimmten benutzerdefinierten Modell verknüpft.
6. Um die Eingabedaten für die Analyse anzugeben, können Sie Text eingeben oder eine Datei hochladen.
  - Um Text einzugeben:
    - a. Wählen Sie „Text eingeben“.
    - b. Geben Sie den Text ein, den Sie analysieren möchten.
  - Um eine Datei hochzuladen:
    - a. Wählen Sie Datei hochladen und geben Sie den Namen der hochzuladenden Datei ein.
    - b. (Optional) Unter Erweiterte Leseaktionen können Sie die Standardaktionen für die Textextraktion überschreiben. Details hierzu finden Sie unter [Festlegen von Optionen für die Textextraktion](#)

Die besten Ergebnisse erzielen Sie, wenn Sie den Typ der Eingabe an den Modelltyp des Klassifikators anpassen. In der Konsole wird eine Warnung angezeigt, wenn Sie ein systemeigenes Dokument an ein Nur-Text-Modell oder Nur-Text-Dokument an ein systemeigenes Dokumentmodell senden. Weitere Informationen finden Sie unter [Modelle zur Trainingsklassifizierung](#).

7. Wählen Sie Analysieren. Amazon Comprehend analysiert die Eingabedaten anhand Ihres benutzerdefinierten Modells. Amazon Comprehend zeigt die erkannten Klassen zusammen mit einer Vertrauensbewertung für jede Klasse an.

## Echtzeitanalyse für benutzerdefinierte Klassifizierung (API)

Sie können die Amazon Comprehend API verwenden, um eine Echtzeitklassifizierung mit einem benutzerdefinierten Modell durchzuführen. Zunächst erstellen Sie einen Endpunkt, um die Echtzeitanalyse auszuführen. Nachdem Sie den Endpunkt erstellt haben, führen Sie die Echtzeitklassifizierung durch.

Die Beispiele in diesem Abschnitt verwenden Befehlsformate für Unix, Linux und macOS. Ersetzen Sie unter Windows den umgekehrten Schrägstrich (\), das Unix-Fortsetzungszeichen, am Ende jeder Zeile durch ein Caret-Zeichen oder Zirkumflex (^).

Informationen zur Bereitstellung des Endpunktdurchsatzes und zu den damit verbundenen Kosten finden Sie unter [Verwenden von Amazon Comprehend-Endpunkten](#).

## Themen

- [Einen Endpunkt für die benutzerdefinierte Klassifizierung erstellen](#)
- [Ausführen einer benutzerdefinierten Klassifizierung in Echtzeit](#)

## Einen Endpunkt für die benutzerdefinierte Klassifizierung erstellen

Das folgende Beispiel zeigt den [CreateEndpoint](#) API-Vorgang unter Verwendung von AWS CLI.

```
aws comprehend create-endpoint \  
  --desired-inference-units number of inference units \  
  --endpoint-name endpoint name \  
  --model-arn arn:aws:comprehend:region:account-id:model/example \  
  --tags Key=My1stTag,Value=Value1
```

Amazon Comprehend antwortet wie folgt:

```
{  
  "EndpointArn": "Arn"  
}
```

## Ausführen einer benutzerdefinierten Klassifizierung in Echtzeit

Nachdem Sie einen Endpunkt für Ihr benutzerdefiniertes Klassifizierungsmodell erstellt haben, verwenden Sie den Endpunkt, um den [ClassifyDocument](#) API-Vorgang auszuführen. Sie können Texteingaben mit dem `bytes` Parameter `text` oder bereitstellen. Geben Sie die anderen Eingabetypen mithilfe des `bytes` Parameters ein.

Bei Bild- und PDF-Dateien können Sie den `DocumentReaderConfig` Parameter verwenden, um die standardmäßigen Textextraktionsaktionen zu überschreiben. Details hierzu finden Sie unter [Festlegen von Optionen für die Textextraktion](#)

Die besten Ergebnisse erzielen Sie, wenn Sie den Typ der Eingabe an den Modelltyp des Klassifikators anpassen. Die API-Antwort enthält eine Warnung, wenn Sie ein

systemeigenes Dokument an ein Nur-Text-Modell oder eine Nur-Text-Datei an ein systemeigenes Dokumentenmodell senden. Weitere Informationen finden Sie unter [Modelle zur Trainingsklassifizierung](#).

Unter Verwendung der AWS Command Line Interface

Die folgenden Beispiele zeigen, wie der CLI-Befehl `classify-document` verwendet wird.

Klassifizieren Sie Text mit dem AWS CLI

Im folgenden Beispiel wird eine Echtzeitklassifizierung für einen Textblock ausgeführt.

```
aws comprehend classify-document \
  --endpoint-arn arn:aws:comprehend:region:account-id:endpoint/endpoint name \
  --text 'From the Tuesday, April 16th, 1912 edition of The Guardian newspaper: The
maiden voyage of the White Star liner Titanic,
the largest ship ever launched ended in disaster. The Titanic started her trip
from Southampton for New York on Wednesday. Late
on Sunday night she struck an iceberg off the Grand Banks of Newfoundland. By
wireless telegraphy she sent out signals of distress,
and several liners were near enough to catch and respond to the call.'
```

Amazon Comprehend antwortet wie folgt:

```
{
  "Classes": [
    {
      "Name": "string",
      "Score": 0.9793661236763
    }
  ]
}
```

Klassifizieren Sie ein halbstrukturiertes Dokument mit dem AWS CLI

Um die benutzerdefinierte Klassifizierung für eine PDF-, Word- oder Bilddatei zu analysieren, führen Sie den `classify-document` Befehl mit der Eingabedatei im `bytes` Parameter aus.

Im folgenden Beispiel wird ein Bild als Eingabedatei verwendet. Es verwendet die `fileb` Option zur Base-64-Kodierung der Bilddatei-Bytes. Weitere Informationen finden Sie unter [Binary large objects](#) im AWS Command Line Interface Benutzerhandbuch.

In diesem Beispiel wird auch eine JSON-Datei mit dem Namen `A` übergeben `config.json`, um die Optionen für die Textextraktion festzulegen.

```
$ aws comprehend classify-document \  
> --endpoint-arn arn \  
> --language-code en \  
> --bytes fileb://image1.jpg \  
> --document-reader-config file://config.json
```

Die Datei `config.json` enthält den folgenden Inhalt.

```
{  
  "DocumentReadMode": "FORCE_DOCUMENT_READ_ACTION",  
  "DocumentReadAction": "EXTRACT_DETECT_DOCUMENT_TEXT"  
}
```

Amazon Comprehend antwortet wie folgt:

```
{  
  "Classes": [  
    {  
      "Name": "string",  
      "Score": 0.9793661236763  
    }  
  ]  
}
```

Weitere Informationen finden Sie [ClassifyDocument](#) in der Amazon Comprehend API-Referenz.

## Ergebnisse für Echtzeitanalysen

### Ausgaben für Texteingaben

Bei Texteingaben enthält die Ausgabe die Liste der Klassen oder Bezeichnungen, die durch die Klassifikatoranalyse identifiziert wurden. Das folgende Beispiel zeigt eine Liste mit zwei Klassen.

```
"Classes": [  
  {
```

```
"Name": "abc",
"Score": 0.2757999897003174,
"Page": 1
},
{
  "Name": "xyz",
  "Score": 0.2721000015735626,
  "Page": 1
}
]
```

## Ausgaben für halbstrukturierte Eingaben

Bei einem halbstrukturierten Eingabedokument oder einer Textdatei kann die Ausgabe die folgenden zusätzlichen Felder enthalten:

- **DocumentMetadata** — Extraktionsinformationen über das Dokument. Die Metadaten enthalten eine Liste von Seiten im Dokument, wobei die Anzahl der Zeichen aus jeder Seite extrahiert wurde. Dieses Feld ist in der Antwort vorhanden, wenn die Anfrage den `Byte` Parameter enthält.
- **DocumentType** — Der Dokumenttyp für jede Seite im Eingabedokument. Dieses Feld ist in der Antwort vorhanden, wenn die Anfrage den `Byte` Parameter enthält.
- **Fehler** — Fehler auf Seitenebene, die das System bei der Verarbeitung des Eingabedokuments erkannt hat. Das Feld ist leer, wenn das System keine Fehler festgestellt hat.
- **Warnungen** — Bei der Verarbeitung des Eingabedokuments wurden Warnungen erkannt. Die Antwort enthält eine Warnung, falls eine Diskrepanz zwischen dem Eingabedokumenttyp und dem Modelltyp besteht, der dem von Ihnen angegebenen Endpunkt zugeordnet ist. Das Feld ist leer, wenn das System keine Warnungen generiert hat.

Weitere Informationen zu diesen Ausgabefeldern finden Sie [ClassifyDocument](#) in der Amazon Comprehend API-Referenz.

Das folgende Beispiel zeigt die Ausgabe für ein einseitiges natives PDF-Eingabedokument.

```
{
  "Classes": [
    {
      "Name": "123",
      "Score": 0.39570000767707825,
      "Page": 1
    }
  ]
}
```

```
    },
    {
      "Name": "abc",
      "Score": 0.2757999897003174,
      "Page": 1
    },
    {
      "Name": "xyz",
      "Score": 0.2721000015735626,
      "Page": 1
    }
  ],
  "DocumentMetadata": {
    "Pages": 1,
    "ExtractedCharacters": [
      {
        "Page": 1,
        "Count": 2013
      }
    ]
  },
  "DocumentType": [
    {
      "Page": 1,
      "Type": "NATIVE_PDF"
    }
  ]
}
```

## Asynchrone Jobs ausführen

Nachdem Sie einen benutzerdefinierten Klassifikator trainiert haben, können Sie asynchrone Jobs verwenden, um große Dokumente oder mehrere Dokumente in einem Stapel zu analysieren.

Die benutzerdefinierte Klassifizierung akzeptiert eine Vielzahl von Eingabedokumenttypen. Details hierzu finden Sie unter [Eingaben für asynchrone benutzerdefinierte Analysen](#).

Wenn Sie Bilddateien oder gescannte PDF-Dokumente analysieren möchten, muss Ihre IAM-Richtlinie die Erlaubnis zur Verwendung von zwei Amazon Textract Textract-API-Methoden (DetectDocumentText und AnalyzeDocument) gewähren. Amazon Comprehend ruft diese Methoden während der Textextraktion auf. Eine Beispielrichtlinie finden Sie unter [Erforderliche Berechtigungen zum Ausführen von Dokumentanalyseaktionen](#).



Verwenden Sie für die Klassifizierung halbstrukturierter Dokumente (Bild-, PDF- oder Docx-Dateien) mithilfe eines Klartextmodells das Eingabeformat `one document per file`. Nehmen Sie den `DocumentReaderConfig` Parameter auch in Ihre Anfrage auf. [StartDocumentClassificationJob](#)

## Themen

- [Dateiformate für die asynchrone Analyse](#)
- [Analyseaufträge für die benutzerdefinierte Klassifizierung \(Konsole\)](#)
- [Analyseaufträge für die benutzerdefinierte Klassifizierung \(API\)](#)
- [Ausgaben für asynchrone Analysejobs](#)

## Dateiformate für die asynchrone Analyse

Wenn Sie eine asynchrone Analyse mit Ihrem Modell ausführen, haben Sie die Wahl zwischen verschiedenen Formaten für Eingabedokumente: `one document per line` oder `one document per file`. Welches Format Sie verwenden, hängt von der Art der Dokumente ab, die Sie analysieren möchten, wie in der folgenden Tabelle beschrieben.

Beschreibung	Format
<p>Die Eingabe enthält mehrere Dateien. Jede Datei enthält ein Eingabedokument. Dieses Format eignet sich am besten für Sammlungen großer Dokumente, wie Zeitungsartikel oder wissenschaftliche Arbeiten.</p> <p>Verwenden Sie dieses Format auch für halbstrukturierte Dokumente (Bild-, PDF- oder Docx-Dateien) mithilfe eines systemeigenen Dokumentenklassifizierers.</p>	Ein Dokument pro Datei
<p>Die Eingabe besteht aus einer oder mehreren Dateien. Jede Zeile in der Datei ist ein separates Eingabedokument. Dieses Format eignet sich am besten für kurze Dokumente wie Textnachrichten oder Beiträge in sozialen Netzwerken.</p>	Ein Dokument pro Zeile

## Ein Dokument pro Datei

Beim `one document per file` Format steht jede Datei für ein Eingabedokument.

## Ein Dokument pro Zeile

Bei diesem `one document per line` Format wird jedes Dokument in einer separaten Zeile platziert und es wird keine Kopfzeile verwendet. Das Etikett ist nicht in jeder Zeile enthalten (da Sie das Etikett für das Dokument noch nicht kennen). Jede Zeile der Datei (das Ende des einzelnen Dokuments) muss mit einem Zeilenvorschub (LF,\n), einem Zeilenwechsel (CR,\r) oder beidem (CRLF,\r\n) enden. Verwenden Sie nicht das UTF-8-Zeilentrennzeichen (u+2028), um eine Zeile zu beenden.

Das folgende Beispiel zeigt das Format der Eingabedatei.

```
Text of document 1 \n
Text of document 2 \n
Text of document 3 \n
Text of document 4 \n
```

Verwenden Sie für beide Formate die UTF-8-Kodierung für Textdateien. Nachdem Sie die Dateien vorbereitet haben, platzieren Sie sie in dem S3-Bucket, den Sie für die Eingabedaten verwenden.

Wenn Sie einen Klassifizierungsjob starten, geben Sie diesen Amazon S3 S3-Speicherort für Ihre Eingabedaten an. Der URI muss sich in derselben Region befinden wie der API-Endpunkt, den Sie aufrufen. Der URI kann auf eine einzelne Datei verweisen (wie bei der Methode „Ein Dokument pro Zeile“), oder er kann das Präfix für eine Sammlung von Datendateien sein.

Wenn Sie beispielsweise den URI `S3://bucketName/prefix` verwenden und das Präfix eine einzelne Datei ist, verwendet Amazon Comprehend diese Datei als Eingabe. Wenn mehr als eine Datei mit dem Präfix beginnt, verwendet Amazon Comprehend sie alle als Eingabe.

Gewähren Sie Amazon Comprehend Zugriff auf den S3-Bucket, der Ihre Dokumentensammlungs- und Ausgabedateien enthält. Weitere Informationen finden Sie unter [Rollenbasierte Berechtigungen, die für asynchrone Operationen erforderlich sind](#).

## Analyseaufträge für die benutzerdefinierte Klassifizierung (Konsole)

Nachdem Sie einen [benutzerdefinierten Dokumentenklassifizierer](#) erstellt und trainiert haben, können Sie die Konsole verwenden, um benutzerdefinierte Klassifizierungsaufträge mit dem Modell auszuführen.

## So erstellen Sie einen benutzerdefinierten Klassifizierungsjob (Konsole)

1. [Melden Sie sich bei der Amazon Comprehend Comprehend-Konsole an AWS Management Console und öffnen Sie sie unter https://console.aws.amazon.com/comprehend/](https://console.aws.amazon.com/comprehend/)
2. Wählen Sie im linken Menü Analyse-Jobs und dann Create Job aus.
3. Geben Sie dem Klassifizierungsjob einen Namen. Der Name muss für Ihr Konto und Ihre aktuelle Region eindeutig sein.
4. Wählen Sie unter Analysetyp die Option Benutzerdefinierte Klassifizierung aus.
5. Wählen Sie unter Klassifikator auswählen den benutzerdefinierten Klassifikator aus, den Sie verwenden möchten.
6. (Optional) Wenn Sie die Daten verschlüsseln möchten, die Amazon Comprehend bei der Verarbeitung Ihres Auftrags verwendet, wählen Sie Auftragsverschlüsselung. Wählen Sie dann, ob Sie einen KMS-Schlüssel verwenden möchten, der mit dem aktuellen Konto verknüpft ist, oder einen von einem anderen Konto.
  - Wenn Sie einen Schlüssel verwenden, der dem aktuellen Konto zugeordnet ist, wählen Sie die Schlüssel-ID für die KMS-Schlüssel-ID.
  - Wenn Sie einen Schlüssel verwenden, der einem anderen Konto zugeordnet ist, geben Sie den ARN für die Schlüssel-ID unter KMS-Schlüssel-ARN ein.


### Note

Weitere Informationen zur Erstellung und Verwendung von KMS-Schlüsseln und der zugehörigen Verschlüsselung finden Sie unter [Schlüsselverwaltungsdienst \(KMS\)](#).

7. Geben Sie unter Eingabedaten den Speicherort des Amazon S3 S3-Buckets ein, der Ihre Eingabedokumente enthält, oder navigieren Sie dorthin, indem Sie Browse S3 wählen. Dieser Bucket muss sich in derselben Region befinden wie die API, die Sie aufrufen. Die IAM-Rolle, die Sie für Zugriffsberechtigungen für den Klassifizierungsjob verwenden, muss über Leseberechtigungen für den S3-Bucket verfügen.

Um ein Höchstmaß an Genauigkeit beim Trainieren eines Modells zu erreichen, müssen Sie den Typ der Eingabe an den Modelltyp des Klassifikators anpassen. Der Classifier-Job gibt eine Warnung zurück, wenn Sie systemeigene Dokumente an ein Nur-Text-Modell oder Nur-Text-Dokumente an ein systemeigenes Dokumentenmodell weiterleiten. Weitere Informationen finden Sie unter [Modelle zur Trainingsklassifizierung](#).

8. (Optional) Unter Eingabeformat können Sie das Format der Eingabedokumente auswählen. Das Format kann ein Dokument pro Datei oder ein Dokument pro Zeile in einer einzigen Datei sein. Ein Dokument pro Zeile gilt nur für Textdokumente.
9. (Optional) Im Lesemodus Dokument können Sie die Standardaktionen zur Textextraktion überschreiben. Weitere Informationen finden Sie unter [Festlegen von Optionen für die Textextraktion](#).
10. Geben Sie unter Ausgabedaten den Speicherort des Amazon S3 S3-Buckets ein, in den Amazon Comprehend die Ausgabedaten des Jobs schreiben soll, oder navigieren Sie zu ihm, indem Sie Browse S3 wählen. Dieser Bucket muss sich in derselben Region befinden wie die API, die Sie aufrufen. Die IAM-Rolle, die Sie für Zugriffsberechtigungen für den Klassifizierungsjob verwenden, muss über Schreibberechtigungen für den S3-Bucket verfügen.
11. (Optional) Wenn Sie das Ausgabeergebnis Ihres Jobs verschlüsseln möchten, wählen Sie Verschlüsselung. Wählen Sie dann aus, ob Sie einen KMS-Schlüssel verwenden möchten, der mit dem aktuellen Konto verknüpft ist, oder einen von einem anderen Konto.
  - Wenn Sie einen Schlüssel verwenden, der dem aktuellen Konto zugeordnet ist, wählen Sie den Schlüsselalias oder die Schlüssel-ID für die KMS-Schlüssel-ID.
  - Wenn Sie einen Schlüssel verwenden, der einem anderen Konto zugeordnet ist, geben Sie den ARN für den Schlüsselalias oder die Schlüssel-ID unter KMS-Schlüssel-ID ein.
12. (Optional) Um Ihre Ressourcen von einer VPC aus in Amazon Comprehend zu starten, geben Sie die VPC-ID unter VPC ein oder wählen Sie die ID aus der Drop-down-Liste aus.
  1. Wählen Sie das Subnetz unter Subnetz (s) aus. Nachdem Sie das erste Subnetz ausgewählt haben, können Sie weitere auswählen.
  2. Wählen Sie unter Sicherheitsgruppe (n) die zu verwendende Sicherheitsgruppe aus, falls Sie eine angegeben haben. Nachdem Sie die erste Sicherheitsgruppe ausgewählt haben, können Sie weitere auswählen.

 Note

Wenn Sie eine VPC mit Ihrem Klassifizierungsjob verwenden, müssen die für die Operationen Create und Start DataAccessRole verwendeten VPC Berechtigungen für die VPC gewähren, die auf den Ausgabe-Bucket zugreift.

13. Wählen Sie Job erstellen, um den Job zur Dokumentenklassifizierung zu erstellen.

## Analyseaufträge für die benutzerdefinierte Klassifizierung (API)

Nachdem Sie einen benutzerdefinierten Dokumentenklassifizierer [erstellt und trainiert](#) haben, können Sie den Klassifikator verwenden, um Analyseaufträge auszuführen.

Verwenden Sie den [StartDocumentClassificationJob](#)Vorgang, um mit der Klassifizierung von Dokumenten ohne Etikett zu beginnen. Sie geben den S3-Bucket an, der die Eingabedokumente enthält, den S3-Bucket für die Ausgabedokumente und den zu verwendenden Klassifikator.

Um ein Höchstmaß an Genauigkeit beim Trainieren eines Modells zu erreichen, müssen Sie den Typ der Eingabe an den Modelltyp des Klassifikators anpassen. Der Classifier-Job gibt eine Warnung zurück, wenn Sie systemeigene Dokumente an ein Nur-Text-Modell oder Nur-Text-Dokumente an ein systemeigenes Dokumentenmodell weiterleiten. Weitere Informationen finden Sie unter [Modelle zur Trainingsklassifizierung](#).

[StartDocumentClassificationJob](#) ist asynchron. Sobald Sie den Job gestartet haben, können Sie den [DescribeDocumentClassificationJob](#)Vorgang verwenden, um seinen Fortschritt zu überwachen.

Wenn das Status Feld in der Antwort angezeigt wird `COMPLETED`, können Sie an dem von Ihnen angegebenen Speicherort auf die Ausgabe zugreifen.

### Themen

- [Mit dem AWS Command Line Interface](#)
- [Verwenden des AWS SDK for Java oder SDK für Python](#)

### Mit dem AWS Command Line Interface

Die folgenden Beispiele für den `StartDocumentClassificationJob` Vorgang und andere benutzerdefinierte Klassifikator-APIs mit dem AWS CLI.

Die folgenden Beispiele verwenden das Befehlsformat für Unix, Linux und macOS. Ersetzen Sie unter Windows den umgekehrten Schrägstrich (`\`), das Unix-Fortsetzungszeichen, am Ende jeder Zeile durch ein Caret-Zeichen oder Zirkumflex (`^`).

Führen Sie mithilfe des `StartDocumentClassificationJob` Vorgangs einen benutzerdefinierten Klassifizierungsauftrag aus.

```
aws comprehend start-document-classification-job \  
  --region region \  
  --input-s3-uri s3://input-bucket/
```

```
--document-classifier-arn arn:aws:comprehend:region:account number:document-  
classifier/testDelete \  
--input-data-config S3Uri=s3://S3Bucket/docclass/file  
name,InputFormat=ONE_DOC_PER_LINE \  
--output-data-config S3Uri=s3://S3Bucket/output \  
--data-access-role-arn arn:aws:iam::account number:role/resource name
```

Rufen Sie mithilfe der `DescribeDocumentClassificationJob` Operation Informationen zu einem benutzerdefinierten Klassifikator mit der Job-ID ab.

```
aws comprehend describe-document-classification-job \  
--region region \  
--job-id job id
```

Listet alle benutzerdefinierten Klassifizierungsaufträge in Ihrem Konto auf, die den `ListDocumentClassificationJobs` Vorgang verwenden.

```
aws comprehend list-document-classification-jobs  
--region region
```

## Verwenden des AWS SDK for Java oder SDK für Python

SDK-Beispiele zum Starten eines benutzerdefinierten Classifier-Jobs finden Sie unter [Verwendung StartDocumentClassificationJob mit einem AWS SDK oder CLI](#).

## Ausgaben für asynchrone Analysejobs

Nach Abschluss eines Analyseauftrags werden die Ergebnisse in dem S3-Bucket gespeichert, den Sie in der Anfrage angegeben haben.

### Ausgaben für Texteingaben

Für jedes Format von Texteingabedokumenten (mehrere Klassen oder mehrere Etiketten) besteht die Jobausgabe aus einer einzigen Datei mit dem Namen `output.tar.gz`. Es handelt sich um eine komprimierte Archivdatei, die eine Textdatei mit der Ausgabe enthält.

### Ausgabe mit mehreren Klassen

Wenn Sie einen Klassifikator verwenden, der im Mehrklassenmodus trainiert wurde, werden Ihre Ergebnisse angezeigt. `classes` Jede dieser Klassen `classes` wird verwendet, um beim Training Ihres Klassifikators eine Reihe von Kategorien zu erstellen.

Weitere Informationen zu diesen Ausgabefeldern finden Sie [ClassifyDocument](#) in der Amazon Comprehend API-Referenz.

In den folgenden Beispielen werden die folgenden Klassen verwendet, die sich gegenseitig ausschließen.

```
DOCUMENTARY
SCIENCE_FICTION
ROMANTIC_COMEDY
SERIOUS_DRAMA
OTHER
```

Wenn Ihr Eingabedatenformat ein Dokument pro Zeile ist, enthält die Ausgabedatei eine Zeile für jede Zeile in der Eingabe. Jede Zeile enthält den Dateinamen, die auf Null basierende Zeilennummer der Eingabezeile und die Klasse oder Klassen, die im Dokument gefunden wurden. Es endet mit der Gewissheit, dass Amazon Comprehend davon überzeugt ist, dass die einzelne Instance korrekt klassifiziert wurde.

Beispielsweise:

```
{"File": "file1.txt", "Line": "0", "Classes": [{"Name": "Documentary", "Score": 0.8642}, {"Name": "Other", "Score": 0.0381}, {"Name": "Serious_Drama", "Score": 0.0372}]}
{"File": "file1.txt", "Line": "1", "Classes": [{"Name": "Science_Fiction", "Score": 0.5}, {"Name": "Science_Fiction", "Score": 0.0381}, {"Name": "Science_Fiction", "Score": 0.0372}]}
{"File": "file2.txt", "Line": "2", "Classes": [{"Name": "Documentary", "Score": 0.1}, {"Name": "Documentary", "Score": 0.0381}, {"Name": "Documentary", "Score": 0.0372}]}
{"File": "file2.txt", "Line": "3", "Classes": [{"Name": "Serious_Drama", "Score": 0.3141}, {"Name": "Other", "Score": 0.0381}, {"Name": "Other", "Score": 0.0372}]}
```

Wenn Ihr Eingabedatenformat ein Dokument pro Datei ist, enthält die Ausgabedatei eine Zeile für jedes Dokument. Jede Zeile enthält den Namen der Datei und die Klasse oder Klassen, die im Dokument gefunden wurden. Es endet mit der Gewissheit, dass Amazon Comprehend die einzelne Instanz korrekt klassifiziert hat.

Beispielsweise:

```
{"File": "file0.txt", "Classes": [{"Name": "Documentary", "Score": 0.8642}, {"Name": "Other", "Score": 0.0381}, {"Name": "Serious_Drama", "Score": 0.0372}]}
```

```
{
  "File": "file1.txt",
  "Classes": [
    { "Name": "Science_Fiction", "Score": 0.5 },
    { "Name": "Science_Fiction", "Score": 0.0381 },
    { "Name": "Science_Fiction", "Score": 0.0372 }
  ]
}
{"File": "file2.txt", "Classes": [{"Name": "Documentary", "Score": 0.1}, {"Name": "Documentary", "Score": 0.0381}, {"Name": "Documentary", "Score": 0.0372}]}
{"File": "file3.txt", "Classes": [{"Name": "Serious_Drama", "Score": 0.3141}, {"Name": "Other", "Score": 0.0381}, {"Name": "Other", "Score": 0.0372}]}
```

## Ausgabe mit mehreren Labels

Wenn Sie einen Klassifikator verwenden, der im Multi-Label-Modus trainiert wurde, werden Ihre Ergebnisse angezeigt. Jedes dieser Labels wird verwendet, um den Satz von Kategorien zu erstellen, wenn Sie Ihren Klassifikator trainieren.

In den folgenden Beispielen werden diese eindeutigen Bezeichnungen verwendet.

```
SCIENCE_FICTION
ACTION
DRAMA
COMEDY
ROMANCE
```

Wenn Ihr Eingabedatenformat ein Dokument pro Zeile ist, enthält die Ausgabedatei eine Zeile für jede Zeile in der Eingabe. Jede Zeile enthält den Dateinamen, die auf Null basierende Zeilennummer der Eingabezeile und die Klasse oder Klassen, die im Dokument gefunden wurden. Es endet mit der Gewissheit, dass Amazon Comprehend davon überzeugt ist, dass die einzelne Instance korrekt klassifiziert wurde.

## Beispielsweise:

```
{
  "File": "file1.txt",
  "Line": "0",
  "Labels": [
    { "Name": "Action", "Score": 0.8642 },
    { "Name": "Drama", "Score": 0.650 },
    { "Name": "Science Fiction", "Score": 0.0372 }
  ]
}
{"File": "file1.txt", "Line": "1", "Labels": [{"Name": "Comedy", "Score": 0.5}, {"Name": "Action", "Score": 0.0381}, {"Name": "Drama", "Score": 0.0372}]}
{"File": "file1.txt", "Line": "2", "Labels": [{"Name": "Action", "Score": 0.9934}, {"Name": "Drama", "Score": 0.0381}, {"Name": "Action", "Score": 0.0372}]}
{"File": "file1.txt", "Line": "3", "Labels": [{"Name": "Romance", "Score": 0.9845}, {"Name": "Comedy", "Score": 0.8756}, {"Name": "Drama", "Score": 0.7723}, {"Name": "Science_Fiction", "Score": 0.6157}]}
```

Wenn Ihr Eingabedatenformat ein Dokument pro Datei ist, enthält die Ausgabedatei eine Zeile für jedes Dokument. Jede Zeile enthält den Namen der Datei und die Klasse oder Klassen, die im



Dokument gefunden wurden. Es endet mit der Gewissheit, dass Amazon Comprehend die einzelne Instanz korrekt klassifiziert hat.

Beispielsweise:

```
{
  "File": "file0.txt",
  "Labels": [
    { "Name": "Action", "Score": 0.8642 },
    { "Name": "Drama", "Score": 0.650 },
    { "Name": "Science Fiction", "Score": 0.0372 }
  ]
},
{
  "File": "file1.txt",
  "Labels": [
    { "Name": "Comedy", "Score": 0.5 },
    { "Name": "Action", "Score": 0.0381 },
    { "Name": "Drama", "Score": 0.0372 }
  ]
},
{
  "File": "file2.txt",
  "Labels": [
    { "Name": "Action", "Score": 0.9934 },
    { "Name": "Drama", "Score": 0.0381 },
    { "Name": "Action", "Score": 0.0372 }
  ]
},
{
  "File": "file3.txt",
  "Labels": [
    { "Name": "Romance", "Score": 0.9845 },
    { "Name": "Comedy", "Score": 0.8756 },
    { "Name": "Drama", "Score": 0.7723 },
    { "Name": "Science_Fiction", "Score": 0.6157 }
  ]
}
```

## Ausgaben für halbstrukturierte Eingabedokumente

Bei halbstrukturierten Eingabedokumenten kann die Ausgabe die folgenden zusätzlichen Felder enthalten:

- **DocumentMetadata** — Extraktionsinformationen über das Dokument. Die Metadaten enthalten eine Liste von Seiten im Dokument, wobei die Anzahl der Zeichen aus jeder Seite extrahiert wurde. Dieses Feld ist in der Antwort vorhanden, wenn die Anfrage den `Byte` Parameter enthält.
- **DocumentType** — Der Dokumenttyp für jede Seite im Eingabedokument. Dieses Feld ist in der Antwort vorhanden, wenn die Anfrage den `Byte` Parameter enthält.
- **Fehler** — Fehler auf Seitenebene, die das System bei der Verarbeitung des Eingabedokuments erkannt hat. Das Feld ist leer, wenn das System keine Fehler festgestellt hat.

Weitere Informationen zu diesen Ausgabefeldern finden Sie [ClassifyDocument](#) in der Amazon Comprehend API-Referenz.

Das folgende Beispiel zeigt die Ausgabe für eine zweiseitige gescannte PDF-Datei.

```
[{ #First page output
  "Classes": [
    {
      "Name": "__label__2 ",
      "Score": 0.9993996620178223
    },
  ],
}
```

```
    {
      "Name": "__label__3 ",
      "Score": 0.0004330444789957255
    }
  ],
  "DocumentMetadata": {
    "PageNumber": 1,
    "Pages": 2
  },
  "DocumentType": "ScannedPDF",
  "File": "file.pdf",
  "Version": "VERSION_NUMBER"
},
#Second page output
{
  "Classes": [
    {
      "Name": "__label__2 ",
      "Score": 0.9993996620178223
    },
    {
      "Name": "__label__3 ",
      "Score": 0.0004330444789957255
    }
  ],
  "DocumentMetadata": {
    "PageNumber": 2,
    "Pages": 2
  },
  "DocumentType": "ScannedPDF",
  "File": "file.pdf",
  "Version": "VERSION_NUMBER"
}]
}}
```

# Erkennung benutzerdefinierter Entitäten

Die Erkennung benutzerdefinierter Entitäten erweitert die Funktion von Amazon Comprehend, indem sie Ihnen hilft, Ihre spezifischen neuen Entitätstypen zu identifizieren, die nicht zu den voreingestellten [generischen Entitätstypen](#) gehören. Das bedeutet, dass Sie Dokumente analysieren und Entitäten wie Produktcodes oder geschäftsspezifische Entitäten extrahieren können, die Ihren speziellen Anforderungen entsprechen.

Die Entwicklung eines genauen benutzerdefinierten Entity Recognizers allein kann ein komplexer Prozess sein, der die Vorbereitung großer Mengen manuell kommentierter Trainingsdokumente und die Auswahl der richtigen Algorithmen und Parameter für das Modelltraining erfordert. Amazon Comprehend trägt dazu bei, die Komplexität zu reduzieren, indem es automatische Anmerkungen und Modellentwicklung bereitstellt, um ein benutzerdefiniertes Entitätserkennungsmodell zu erstellen.

Das Erstellen eines benutzerdefinierten Entitätserkennungsmodells ist ein effektiverer Ansatz als die Verwendung von Zeichenfolgenabgleich oder regulären Ausdrücken zum Extrahieren von Entitäten aus Dokumenten. Um beispielsweise ENGINEER-Namen in einem Dokument zu extrahieren, ist es schwierig, alle möglichen Namen aufzuzählen. Darüber hinaus ist es ohne Kontext schwierig, zwischen ENGINEER-Namen und ANALYST-Namen zu unterscheiden. Ein benutzerdefiniertes Entitätserkennungsmodell kann den Kontext lernen, in dem diese Namen wahrscheinlich erscheinen. Darüber hinaus erkennt der Zeichenfolgenabgleich keine Entitäten, die Tippfehler haben oder neuen Namenskonventionen folgen, während dies mit einem benutzerdefinierten Modell möglich ist.

Sie haben zwei Möglichkeiten, ein benutzerdefiniertes Modell zu erstellen:

1. Anmerkungen – Stellen Sie einen Datensatz bereit, der kommentierte Entitäten für das Modelltraining enthält.
2. Entitätslisten (nur Klartext) – Stellen Sie eine Liste von Entitäten und deren Typbezeichnung bereit (z. B. PRODUCT\_CODES und eine Reihe von unbemerkten Dokumenten, die diese Entitäten für das Modelltraining enthalten).

Wenn Sie einen benutzerdefinierten Entity Recognizer mit kommentierten PDF-Dateien erstellen, können Sie diesen Recognizer mit einer Vielzahl von Eingabedateiformaten verwenden: Klartext, Bilddateien (JPG, PNG, TIFF), PDF-Dateien und Word-Dokumente, ohne dass eine Vorverarbeitung oder eine Dokumentabflachung erforderlich ist. Amazon Comprehend unterstützt keine Anmerkung von Bilddateien oder Word-Dokumenten.

**Note**

Ein benutzerdefinierter Entity Recognizer, der kommentierte PDF-Dateien verwendet, unterstützt nur englische Dokumente.

Sie können ein Modell mit bis zu 25 benutzerdefinierten Entitäten gleichzeitig trainieren. Weitere Informationen finden Sie auf der [Seite Richtlinien und Kontingente](#).

Nachdem Ihr Modell trainiert wurde, können Sie das Modell für die Erkennung von Entitäten in Echtzeit und in Aufträgen zur Entitätserkennung verwenden.

**Themen**

- [Vorbereiten von Trainingsdaten für Entity Recognizer](#)
- [Trainieren von benutzerdefinierten Entity Recognizer-Modellen](#)
- [Ausführen einer benutzerdefinierten Erkennungsanalyse in Echtzeit](#)
- [Ausführen von Analyseaufträgen zur Erkennung benutzerdefinierter Entitäten](#)

## Vorbereiten von Trainingsdaten für Entity Recognizer

Um ein erfolgreiches benutzerdefiniertes Entitätserkennungsmodell zu trainieren, ist es wichtig, dem Modell-Planer qualitativ hochwertige Daten als Eingabe zur Verfügung zu stellen. Ohne gute Daten lernt das Modell nicht, wie Entitäten korrekt identifiziert werden.

Sie können eine von zwei Möglichkeiten wählen, Daten für Amazon Comprehend bereitzustellen, um ein benutzerdefiniertes Entitätserkennungsmodell zu trainieren:

- **Entitätsliste** – Listet die spezifischen Entitäten auf, damit Amazon Comprehend trainieren kann, Ihre benutzerdefinierten Entitäten zu identifizieren. Hinweis: Entitätslisten können nur für Klartextdokumente verwendet werden.
- **Anmerkungen** – Gibt den Standort Ihrer Entitäten in einer Reihe von Dokumenten an, sodass Amazon Comprehend sowohl für die Entität als auch für ihren Kontext trainieren kann. Um ein Modell für die Analyse von Bilddateien, PDFs oder Word-Dokumenten zu erstellen, müssen Sie Ihren Recognizer mithilfe von PDF-Anmerkungen trainieren.

In beiden Fällen lernt Amazon Comprehend die Art der Dokumente und den Kontext kennen, in dem die Entitäten auftreten, und erstellt einen Erkenner, der verallgemeinert werden kann, um die neuen Entitäten bei der Analyse von Dokumenten zu erkennen.

Wenn Sie ein benutzerdefiniertes Modell erstellen (oder eine neue Version trainieren), können Sie einen Testdatensatz bereitstellen. Wenn Sie keine Testdaten angeben, reserviert Amazon Comprehend 10 % der Eingabedokumente, um das Modell zu testen. Amazon Comprehend trainiert das Modell mit den verbleibenden Dokumenten.

Wenn Sie einen Testdatensatz für Ihren Trainingssatz für Anmerkungen bereitstellen, müssen die Testdaten mindestens eine Anmerkung für jeden der in der Erstellungsanforderung angegebenen Entitätstypen enthalten.

## Themen

- [Wann sollten Anmerkungen im Vergleich zu Entitätslisten verwendet werden?](#)
- [Entitätslisten \(nur Klartext\)](#)
- [Anmerkungen](#)

## Wann sollten Anmerkungen im Vergleich zu Entitätslisten verwendet werden?

Das Erstellen von Anmerkungen erfordert mehr Arbeit als das Erstellen einer Entitätsliste, aber das resultierende Modell kann deutlich genauer sein. Die Verwendung einer Entitätsliste ist schneller und weniger arbeitsintensiv, aber die Ergebnisse werden weniger verfeinert und weniger genau. Dies liegt daran, dass die Anmerkungen mehr Kontext bieten, den Amazon Comprehend beim Training des Modells verwenden soll. Ohne diesen Kontext weist Amazon Comprehend eine höhere Anzahl falsch positiver Ergebnisse auf, wenn versucht wird, die Entitäten zu identifizieren.

Es gibt Szenarien, in denen es aus geschäftlichen Gründen sinnvoller ist, die höheren Kosten und den Workload durch die Verwendung von Anmerkungen zu vermeiden. Der Name John ist beispielsweise für Ihre Suche von Bedeutung, aber ob es sich um die genaue Person handelt, ist nicht relevant. Oder die Metriken bei Verwendung der Entitätsliste sind gut genug, um Ihnen die von Ihnen benötigten Erkennungsergebnisse zu liefern. In solchen Fällen kann die Verwendung einer Entitätsliste stattdessen die effektivere Wahl sein.

Wir empfehlen in den folgenden Fällen die Verwendung des Annotationsmodus:

- Wenn Sie Inferenzen für Bilddateien, PDFs oder Word-Dokumente ausführen möchten. In diesem Szenario trainieren Sie ein Modell mit kommentierten PDF-Dateien und verwenden das Modell, um Inferenzaufträge für Bilddateien, PDFs und Word-Dokumente auszuführen.
- Wenn die Bedeutung der Entitäten mehrdeutig und kontextabhängig sein könnte. Der Begriff Amazon könnte sich beispielsweise entweder auf den Arzt in Brasilien oder den Online-Shop-Anbieter Amazon.com beziehen. Wenn Sie einen benutzerdefinierten Entity Recognizer erstellen, um Geschäftseinheiten wie Amazon zu identifizieren, sollten Sie Anmerkungen anstelle einer Entitätsliste verwenden, da diese Methode besser in der Lage ist, Kontext zu verwenden, um Entitäten zu finden.
- Wenn Sie mit der Einrichtung eines Prozesses zum Abrufen von Anmerkungen vertraut sind, kann dies einen gewissen Aufwand erfordern.

In den folgenden Fällen empfehlen wir die Verwendung einer Entitätsliste:

- Wenn Sie bereits über eine Liste von Entitäten verfügen oder wenn es relativ einfach ist, eine umfassende Liste von Entitäten zu erstellen. Wenn Sie eine Entitätsliste verwenden, sollte die Liste vollständig sein oder zumindest die meisten gültigen Entitäten abdecken, die möglicherweise in den Dokumenten erscheinen, die Sie für die Schulung bereitstellen.
- Für Erstbenutzer wird im Allgemeinen empfohlen, eine Entitätsliste zu verwenden, da dies einen geringeren Aufwand erfordert als das Erstellen von Anmerkungen. Beachten Sie jedoch, dass das trainierte Modell möglicherweise nicht so genau ist, wie wenn Sie Anmerkungen verwenden würden.

## Entitätslisten (nur Klartext)

Um ein Modell mithilfe einer Entitätsliste zu trainieren, stellen Sie zwei Informationen bereit: eine Liste der Entitätsnamen mit den entsprechenden benutzerdefinierten Entitätstypen und eine Sammlung von unbemerkten Dokumenten, in denen Ihre Entitäten erscheinen sollen.

Wenn Sie eine Entitätsliste bereitstellen, verwendet Amazon Comprehend einen intelligenteren Algorithmus, um Vorkommen der Entität in den Dokumenten zu erkennen, die als Grundlage für das Training des benutzerdefinierten Entity Recognizer-Modells dienen.

Geben Sie für Entitätslisten mindestens 25 Entitätsübereinstimmungen pro Entitätstyp in der Entitätsliste an.

Eine Entitätsliste für die Erkennung benutzerdefinierter Entitäten benötigt eine CSV-Datei (durch Kommas getrennte Werte) mit den folgenden Spalten:

- Text – Der Text eines Einstiegsbeispiels, genau wie er im Korpus des zugehörigen Dokuments zu sehen ist.
- Typ — Der vom Kunden definierte Entitätstyp. Entitätstypen müssen eine durch Großbuchstaben und Unterstriche getrennte Zeichenfolge wie `MANAGER` oder `SENIOR_MANAGER` enthalten. Pro Modell können bis zu 25 Entitätstypen trainiert werden.

Die Datei `documents.txt` enthält vier Zeilen:

```
Jo Brown is an engineer in the high tech industry.  
John Doe has been a engineer for 14 years.  
Emilio Johnson is a judge on the Washington Supreme Court.  
Our latest new employee, Jane Smith, has been a manager in the industry for 4 years.
```

Die CSV-Datei mit der Liste der Entitäten hat die folgenden Zeilen:

```
Text, Type  
Jo Brown, ENGINEER  
John Doe, ENGINEER  
Jane Smith, MANAGER
```

#### Note

In der Entitätsliste ist der Eintrag für Emilio nicht vorhanden, da er weder die Entität `ENGINEER` noch `MANAGER` enthält.

## Erstellen Ihrer Datendateien

Es ist wichtig, dass sich Ihre Entitätsliste in einer ordnungsgemäß konfigurierten CSV-Datei befindet, damit die Wahrscheinlichkeit von Problemen mit Ihrer Entitätslistendatei minimal ist. Um Ihre CSV-Datei manuell zu konfigurieren, muss Folgendes zutreffen:

- Die UTF-8-Kodierung muss explizit angegeben werden, auch wenn sie in den meisten Fällen als Standard verwendet wird.
- Er muss die Spaltennamen `Type` und `enthaltenText`.

Es wird dringend empfohlen, CSV-Eingabedateien programmgesteuert zu generieren, um potenzielle Probleme zu vermeiden.

Im folgenden Beispiel wird Python verwendet, um ein CSV für die oben gezeigten Anmerkungen zu generieren:

```
import csv
with open("./entitylist/entitylist.csv", "w", encoding="utf-8") as csv_file:
    csv_writer = csv.writer(csv_file)
    csv_writer.writerow(["Text", "Type"])
    csv_writer.writerow(["Jo Brown", "ENGINEER"])
    csv_writer.writerow(["John Doe", "ENGINEER"])
    csv_writer.writerow(["Jane Smith", "MANAGER"])
```

## Bewährte Methoden

Es gibt eine Reihe von Dingen, die Sie berücksichtigen sollten, um bei der Verwendung einer Entitätsliste das beste Ergebnis zu erzielen, darunter:

- Die Reihenfolge der Entitäten in Ihrer Liste hat keine Auswirkungen auf das Modelltraining.
- Verwenden Sie Entitätslistenelemente, die 80 % bis 100 % der positiven Entitätsbeispiele abdecken, die im nicht kommentierten Korpus von Dokumenten erwähnt werden.
- Vermeiden Sie Entitätsbeispiele, die Nicht-Entitäten im Dokumentkorpus entsprechen, indem Sie allgemeine Wörter und Wortgruppen entfernen. Selbst eine Handvoll falscher Übereinstimmungen kann die Genauigkeit Ihres resultierenden Modells erheblich beeinflussen. Ein Wort wie das in der Entitätsliste führt beispielsweise zu einer hohen Anzahl von Übereinstimmungen, bei denen es sich wahrscheinlich nicht um die Entitäten handelt, nach denen Sie suchen, und die sich daher erheblich auf Ihre Genauigkeit auswirken.
- Eingabedaten sollten keine Duplikate enthalten. Das Vorhandensein von doppelten Stichproben kann zu Testsatzbereinigungen führen und sich daher negativ auf den Trainingsprozess, die Modellmetriken und das Verhalten auswirken.
- Stellen Sie Dokumente, die echten Anwendungsfällen ähneln, so genau wie möglich bereit. Verwenden Sie keine Toy-Daten oder synthetisierte Daten für Produktionssysteme. Die Eingabedaten sollten so unterschiedlich wie möglich sein, um Überanpassungen zu vermeiden und das zugrunde liegende Modell anhand realer Beispiele besser verallgemeinern zu können.
- Bei der Entitätsliste wird zwischen Groß- und Kleinschreibung unterschieden und reguläre Ausdrücke werden derzeit nicht unterstützt. Das trainierte Modell kann Entitäten jedoch oft immer



noch erkennen, auch wenn sie nicht genau mit der Groß- und Kleinschreibung übereinstimmen, die in der Entitätsliste angegeben ist.

- Wenn Sie über eine Entität verfügen, die eine Teilzeichenfolge einer anderen Entität ist (z. B. „Smith“ und „JaneISS“, geben Sie beide in der Entitätsliste an.

Weitere Vorschläge finden Sie unter [Verbesserung der Leistung des benutzerdefinierten Entity Recognizers](#)

## Anmerkungen

Anmerkungen kennzeichnen Entitäten im Kontext, indem sie Ihre benutzerdefinierten Entitätstypen den Speicherorten zuordnen, an denen sie in Ihren Trainingsdokumenten vorkommen.

Indem Sie eine Anmerkung zusammen mit Ihren Dokumenten einreichen, können Sie die Genauigkeit des Modells erhöhen. Mit Annotationen geben Sie nicht nur den Speicherort der gesuchten Entität an, sondern bieten auch einen genaueren Kontext zu der gesuchten benutzerdefinierten Entität.

Wenn Sie beispielsweise nach dem Namen John mit dem Entitätstyp JUDGE suchen, kann die Bereitstellung Ihrer Anmerkung dem Modell helfen zu erfahren, dass die Person, die Sie finden möchten, ein Jurator ist. Wenn der Kontext verwendet werden kann, findet Amazon Comprehend keine Personen mit dem Namen John, die Arzt oder Zeugen sind. Ohne Anmerkungen erstellt Amazon Comprehend eine eigene Version einer Anmerkung, ist aber nicht so effektiv bei der Einbeziehung nur von Prüfern. Die Bereitstellung eigener Anmerkungen kann dazu beitragen, bessere Ergebnisse zu erzielen und Modelle zu generieren, die den Kontext beim Extrahieren benutzerdefinierter Entitäten besser nutzen können.

### Themen

- [Mindestanzahl von Anmerkungen](#)
- [Bewährte Methoden für Anmerkungen](#)
- [Klartext-Anmerkungsdateien](#)
- [PDF-Anmerkungsdateien](#)
- [Kommentieren von PDF-Dateien](#)

## Mindestanzahl von Anmerkungen

Die Mindestanzahl der Eingabedokumente und Anmerkungen, die zum Trainieren eines Modells erforderlich sind, hängt von der Art der Anmerkungen ab.

## PDF-Anmerkungen

Um ein Modell für die Analyse von Bilddateien, PDFs oder Word-Dokumenten zu erstellen, trainieren Sie Ihren Recognizer mithilfe von PDF-Anmerkungen. Geben Sie für PDF-Anmerkungen mindestens 250 Eingabedokumente und mindestens 100 Anmerkungen pro Entität an.

Wenn Sie einen Testdatensatz bereitstellen, müssen die Testdaten mindestens eine Anmerkung für jeden der in der Erstellungsanforderung angegebenen Entitätstypen enthalten.

## Klartext-Anmerkungen

Um ein Modell für die Analyse von Textdokumenten zu erstellen, können Sie Ihren Recognizer mithilfe von Klartextanmerkungen trainieren.

Geben Sie für Klartext-Anmerkungen mindestens drei kommentierte Eingabedokumente und mindestens 25 Anmerkungen pro Entität an. Wenn Sie insgesamt weniger als 50 Anmerkungen angeben, reserviert Amazon Comprehend mehr als 10 % der Eingabedokumente, um das Modell zu testen (es sei denn, Sie haben in der Trainingsanforderung einen Testdatensatz angegeben). Denken Sie daran, dass die minimale Dokumentkorpusgröße 5 KB beträgt.

Wenn Ihre Eingabe nur wenige Trainingsdokumente enthält, kann es zu einem Fehler kommen, dass die Trainingseingabedaten zu wenige Dokumente enthalten, die eine der Entitäten erwähnt. Senden Sie den Auftrag erneut mit zusätzlichen Dokumenten, in denen die Entität erwähnt wird.

Wenn Sie einen Testdatensatz bereitstellen, müssen die Testdaten mindestens eine Anmerkung für jeden der in der Erstellungsanforderung angegebenen Entitätstypen enthalten.

Ein Beispiel für das Benchmarking eines Modells mit einem kleinen Datensatz finden Sie unter [Amazon Comprehend kündigt niedrigere Anmerkungslimits für die Erkennung benutzerdefinierter Entitäten](#) an auf der -AWSBlog-Website.

## Bewährte Methoden für Anmerkungen

Bei der Verwendung von Anmerkungen müssen Sie eine Reihe von Dingen berücksichtigen, um das beste Ergebnis zu erzielen, darunter:

- Kommentieren Sie Ihre Daten mit Bedacht und stellen Sie sicher, dass Sie jede Erwähnung der Entität kommentieren. Unpräzise Anmerkungen können zu schlechten Ergebnissen führen.
- Eingabedaten sollten keine Duplikate enthalten, z. B. ein Duplikat eines PDF, das Sie kommentieren möchten. Das Vorhandensein einer doppelten Stichprobe kann zu einer

Testatzbereinigung führen und sich negativ auf den Trainingsprozess, die Modellmetriken und das Modellverhalten auswirken.

- Stellen Sie sicher, dass alle Ihre Dokumente mit Anmerkungen versehen sind und dass die Dokumente ohne Anmerkungen auf fehlende legitime Entitäten zurückzuführen sind, nicht auf steile. Wenn Sie beispielsweise ein Dokument haben, das besagt, dass „J Doe seit 14 Jahren Techniker ist“, sollten Sie auch eine Anmerkung für „J Doe“ sowie „John Doe“ angeben. Andernfalls wird das Modell verwirrt und das Modell erkennt „J Doe“ nicht als ENGINEER. Dies sollte innerhalb desselben Dokuments und dokumentübergreifend konsistent sein.
- Im Allgemeinen führen mehr Anmerkungen zu besseren Ergebnissen.
- Sie können ein Modell mit der [minimalen Anzahl](#) von Dokumenten und Anmerkungen trainieren, aber das Hinzufügen von Daten verbessert normalerweise das Modell. Wir empfehlen, das Volumen annotierter Daten um 10 % zu erhöhen, um die Genauigkeit des Modells zu erhöhen. Sie können Inferenzen für einen Testdatensatz ausführen, der unverändert bleibt und von verschiedenen Modellversionen getestet werden kann. Anschließend können Sie die Metriken für aufeinanderfolgende Modellversionen vergleichen.
- Stellen Sie Dokumente, die echten Anwendungsfällen ähneln, so genau wie möglich bereit. Synthesierte Daten mit sich wiederholenden Mustern sollten vermieden werden. Die Eingabedaten sollten so unterschiedlich wie möglich sein, um Überanpassungen zu vermeiden und dem zugrunde liegenden Modell zu helfen, anhand realer Beispiele besser zu verallgemeinern.
- Es ist wichtig, dass Dokumente in Bezug auf die Anzahl der Wörter unterschiedlich sein sollten. Wenn beispielsweise alle Dokumente in den Trainingsdaten kurz sind, kann das resultierende Modell Schwierigkeiten haben, Entitäten in längeren Dokumenten vorherzusagen.
- Versuchen Sie, dieselbe Datenverteilung für das Training bereitzustellen, die Sie voraussichtlich verwenden, wenn Sie Ihre benutzerdefinierten Entitäten tatsächlich erkennen (Inferenzzeit). Wenn Sie beispielsweise zum Zeitpunkt der Inferenz erwarten, uns Dokumente zu senden, die keine Entitäten enthalten, sollte dies auch Teil Ihres Trainingsdokumentsatzes sein.

Weitere Vorschläge finden Sie unter [Verbesserung der Leistung von benutzerdefinierten Entitätserkennungen](#).

## Klartext-Anmerkungsdateien

Für Klartext-Anmerkungen erstellen Sie eine CSV-Datei (durch Kommas getrennte Werte), die eine Liste von Anmerkungen enthält. Die CSV-Datei muss die folgenden Spalten enthalten, wenn Ihr Eingabeformat für die Trainingsdatei ein Dokument pro Zeile ist.

Datei	Linien	Offset beginnen	Endversatz	Typ
Der Name der Datei, die das Dokument enthält. Wenn sich beispielsweise eine der Dokumentdateien unter <code>my-S3-bucket/test-documents.txt</code> befindet, lautet der Wert in der File Spalte <code>documents.txt</code> . Sie müssen die Dateierweiterung (in diesem Fall <code>.txt</code> ) in den Dateinamen aufnehmen.	Die Zeilennummer, die die Entität enthält. Lassen Sie diese Spalte weg, wenn Ihr Eingabeformat ein Dokument pro Datei ist.	Der Zeichenversatz im Eingabetext (relativ zum Anfang der Zeile), der anzeigt, wo die Entität beginnt. Das erste Zeichen befindet sich an Position 0.	Der Zeichenversatz im Eingabetext, der anzeigt, wo die Entität endet.	Der vom Kunden definierte Entitätstyp. Entitätstypen müssen eine durch Unterstriche getrennte Zeichenfolge in Großbuchstaben sein. Wir empfehlen die Verwendung beschreibender Entitätstypen wie <code>MANAGER</code> , <code>SENIOR_MANAGER</code> oder <code>PRODUCT_CODE</code> . Pro Modell können bis zu 25 Entitätstypen trainiert werden.

Wenn Ihr Eingabeformat für die Trainingsdatei ein Dokument pro Datei ist, lassen Sie die Zeilennummernspalte weg und die Werte Offset beginnen und Offset beenden sind die Offsets der Entität vom Anfang des Dokuments.

Das folgende Beispiel gilt für ein Dokument pro Zeile. Die Datei `documents.txt` enthält vier Zeilen (Zeilen 0, 1, 2 und 3):

```
Diego Ramirez is an engineer in the high tech industry.
Emilio Johnson has been an engineer for 14 years.
```

```
J Doe is a judge on the Washington Supreme Court.  
Our latest new employee, Mateo Jackson, has been a manager in the industry for 4 years.
```

Die CSV-Datei mit der Liste der Anmerkungen lautet wie folgt:

```
File, Line, Begin Offset, End Offset, Type  
documents.txt, 0, 0, 13, ENGINEER  
documents.txt, 1, 0, 14, ENGINEER  
documents.txt, 3, 25, 38, MANAGER
```

### Note

In der Annotationsdatei beginnt die Zeilennummer, die die Entität enthält, mit Zeile 0. In diesem Beispiel enthält die CSV-Datei keinen Eintrag für Zeile 2, da in Zeile 2 von keine Entität vorhanden ist `documents.txt`.

## Erstellen Ihrer Datendateien

Es ist wichtig, Ihre Anmerkungen in einer ordnungsgemäß konfigurierten CSV-Datei abzulegen, um das Fehlerrisiko zu verringern. Um Ihre CSV-Datei manuell zu konfigurieren, muss Folgendes zutreffen:

- Die UTF-8-Kodierung muss explizit angegeben werden, auch wenn sie in den meisten Fällen als Standard verwendet wird.
- Die erste Zeile enthält die Spaltenüberschriften: `File`, `Line` (optional), `Begin Offset`, `End Offset`, `Type`.

Es wird dringend empfohlen, die CSV-Eingabedateien programmgesteuert zu generieren, um potenzielle Probleme zu vermeiden.

Im folgenden Beispiel wird Python verwendet, um ein CSV für die zuvor gezeigten Anmerkungen zu generieren:

```
import csv  
with open("./annotations/annotations.csv", "w", encoding="utf-8") as csv_file:  
    csv_writer = csv.writer(csv_file)  
    csv_writer.writerow(["File", "Line", "Begin Offset", "End Offset", "Type"])
```

```
csv_writer.writerow(["documents.txt", 0, 0, 11, "ENGINEER"])
csv_writer.writerow(["documents.txt", 1, 0, 5, "ENGINEER"])
csv_writer.writerow(["documents.txt", 3, 25, 30, "MANAGER"])
```

## PDF-Anmerkungsdateien

Für PDF-Anmerkungen verwenden Sie SageMaker Ground Truth, um einen beschrifteten Datensatz in einer erweiterten Manifestdatei zu erstellen. Ground Truth ist ein Datenbeschriftungsservice, der Ihnen (oder einer Arbeitskraft, die Sie einsetzen) hilft, Trainingsdatensätze für Machine-Learning-Modelle zu erstellen. Amazon Comprehend akzeptiert erweiterte Manifestdateien als Trainingsdaten für benutzerdefinierte Modelle. Sie können diese Dateien bereitstellen, wenn Sie einen benutzerdefinierten Entity Recognizer mithilfe der Amazon Comprehend-Konsole oder der [CreateEntityRecognizer](#) -API-Aktion erstellen.

Sie können den integrierten Ground Truth-Aufgabentyp Named Entity Recognize verwenden, um einen Kennzeichnungsauftrag zu erstellen, damit Auftragnehmer Entitäten im Text identifizieren können. Weitere Informationen finden Sie unter [Erkennung benannter Entitäten](#) im Amazon-SageMaker Entwicklerhandbuch. Weitere Informationen zu Amazon SageMaker Ground Truth finden Sie unter [Verwenden von Amazon SageMaker Ground Truth zum Beschriften von Daten](#).

### Note

Mit Ground Truth können Sie überlappende Labels definieren (Text, den Sie mehreren Labels zuordnen). Die Amazon Comprehend-Entitätserkennung unterstützt jedoch keine überlappenden Labels.

Erweiterte Manifestdateien liegen im JSON-Zeilenformat vor. In diesen Dateien ist jede Zeile ein vollständiges JSON-Objekt, das ein Trainingsdokument und die zugehörigen Labels enthält. Das folgende Beispiel ist eine erweiterte Manifestdatei, die eine Entitätserkennung darin trainiert, die Fähigkeiten von Personen zu erkennen, die im Text erwähnt werden:

```
{"source":"Diego Ramirez is an engineer in the high tech
industry.,"NamedEntityRecognitionDemo":{"annotations":{"entities":
[{"endOffset":13,"startOffset":0,"label":"ENGINEER"}],"labels":
[{"label":"ENGINEER"}]}}, "NamedEntityRecognitionDemo-metadata":
{"entities":[{"confidence":0.92}],"job-name":"labeling-job/
namedentityrecognitiondemo","type":"groundtruth/text-span","creation-
date":"2020-05-14T21:45:27.175903","human-annotated":"yes"}}
```

```

{"source": "J Doe is a judge on the Washington Supreme
  Court.", "NamedEntityRecognitionDemo": {"annotations": {"entities":
[{"endOffset": 5, "startOffset": 0, "label": "JUDGE"}], "labels":
[{"label": "JUDGE"}]}}, "NamedEntityRecognitionDemo-metadata":
{"entities": [{"confidence": 0.72}], "job-name": "labeling-job/
namedentityrecognitiondemo", "type": "groundtruth/text-span", "creation-
date": "2020-05-14T21:45:27.174910", "human-annotated": "yes"}}
{"source": "Our latest new employee, Mateo Jackson, has been a manager in
  the industry for 4 years.", "NamedEntityRecognitionDemo": {"annotations":
{"entities": [{"endOffset": 38, "startOffset": 26, "label": "MANAGER"}], "labels":
[{"label": "MANAGER"}]}}, "NamedEntityRecognitionDemo-metadata":
{"entities": [{"confidence": 0.91}], "job-name": "labeling-job/
namedentityrecognitiondemo", "type": "groundtruth/text-span", "creation-
date": "2020-05-14T21:45:27.174035", "human-annotated": "yes"}}

```

Jede Zeile in dieser JSON-Zeilendatei ist ein vollständiges JSON-Objekt, wobei die Attribute den Dokumenttext, die Anmerkungen und andere Metadaten aus Ground Truth enthalten. Das folgende Beispiel ist ein einzelnes JSON-Objekt in der erweiterten Manifestdatei, aber aus Gründen der Lesbarkeit formatiert:

```

{
  "source": "Diego Ramirez is an engineer in the high tech industry.",
  "NamedEntityRecognitionDemo": {
    "annotations": {
      "entities": [
        {
          "endOffset": 13,
          "startOffset": 0,
          "label": "ENGINEER"
        }
      ],
      "labels": [
        {
          "label": "ENGINEER"
        }
      ]
    }
  },
  "NamedEntityRecognitionDemo-metadata": {
    "entities": [
      {
        "confidence": 0.92
      }
    ]
  }
}

```

```
    ],  
    "job-name": "labeling-job/namedentityrecognitiondemo",  
    "type": "groundtruth/text-span",  
    "creation-date": "2020-05-14T21:45:27.175903",  
    "human-annotated": "yes"  
  }  
}
```

In diesem Beispiel stellt das `source` Attribut den Text des Trainingsdokuments bereit und das `NamedEntityRecognitionDemo` Attribut stellt die Anmerkungen für die Entitäten im Text bereit. Der Name des `NamedEntityRecognitionDemo` Attributs ist beliebig und Sie geben einen Namen Ihrer Wahl an, wenn Sie den Kennzeichnungsauftrag in Ground Truth definieren.

In diesem Beispiel ist das `NamedEntityRecognitionDemo` Attribut der Name des Beschriftungsattributs. Dabei handelt es sich um das Attribut, das die Beschriftungen bereitstellt, die ein Ground Truth-Worker den Trainingsdaten zuweist. Wenn Sie Ihre Trainingsdaten für Amazon Comprehend bereitstellen, müssen Sie einen oder mehrere Bezeichnungsattributnamen angeben. Die Anzahl der von Ihnen angegebenen Attributnamen hängt davon ab, ob Ihre erweiterte Manifestdatei die Ausgabe eines einzelnen Kennzeichnungsauftrags oder eines verketteten Kennzeichnungsauftrags ist.

Wenn Ihre Datei die Ausgabe eines einzelnen Kennzeichnungsauftrags ist, geben Sie den einzelnen Kennzeichnungsattributnamen an, der beim Erstellen des Auftrags in Ground Truth verwendet wurde.

Wenn Ihre Datei die Ausgabe eines verketteten Kennzeichnungsauftrags ist, geben Sie den Kennzeichnungsattributnamen für einen oder mehrere Aufträge in der Kette an. Jeder Name des Beschriftungsattributs stellt die Anmerkungen aus einem einzelnen Auftrag bereit. Sie können bis zu 5 dieser Attribute für erweiterte Manifestdateien angeben, die von verketteten Kennzeichnungsaufträgen erstellt werden.

In einer erweiterten Manifestdatei folgt der Name des Beschriftungsattributs normalerweise dem `source` Schlüssel. Wenn die Datei die Ausgabe eines verketteten Auftrags ist, gibt es mehrere Bezeichnungsattributnamen. Wenn Sie Ihre Trainingsdaten für Amazon Comprehend bereitstellen, geben Sie nur die Attribute an, die Anmerkungen enthalten, die für Ihr Modell relevant sind. Geben Sie nicht die Attribute an, die mit „-metadata“ enden.

Weitere Informationen zu verketteten Kennzeichnungsaufträgen und Beispiele für die Ausgabe, die sie erzeugen, finden Sie unter [Kennzeichnungsaufträge verketteten](#) im Amazon- SageMaker Entwicklerhandbuch.



## Kommentieren von PDF-Dateien

Bevor Sie Ihre Trainings-PDFs in SageMaker Ground Truth kommentieren können, müssen Sie die folgenden Voraussetzungen erfüllen:

- Installieren Sie python3.8.x
- [Installiere jq](#)
- Installieren Sie die [AWS CLI](#)

Wenn Sie die Region us-east-1 verwenden, können Sie die Installation der AWS CLI überspringen, da sie bereits in Ihrer Python-Umgebung installiert ist. In diesem Fall erstellen Sie eine virtuelle Umgebung, um Python 3.8 in AWS Cloud9 zu verwenden.

- [Konfigurieren Sie Ihre Anmeldedaten AWS](#)
- Stellen Sie eine private [SageMaker Ground-Truth-Belegschaft](#) zur Unterstützung von Annotationen zusammen

Notieren Sie sich unbedingt den Namen des Arbeitsteams, den Sie für Ihre neue private Belegschaft wählen, so wie Sie ihn bei der Installation verwenden.

### Themen

- [Einrichten Ihrer Umgebung](#)
- [Eine PDF-Datei in einen S3-Bucket hochladen](#)
- [Einen Annotationsjob erstellen](#)
- [Kommentieren mit SageMaker Ground Truth](#)

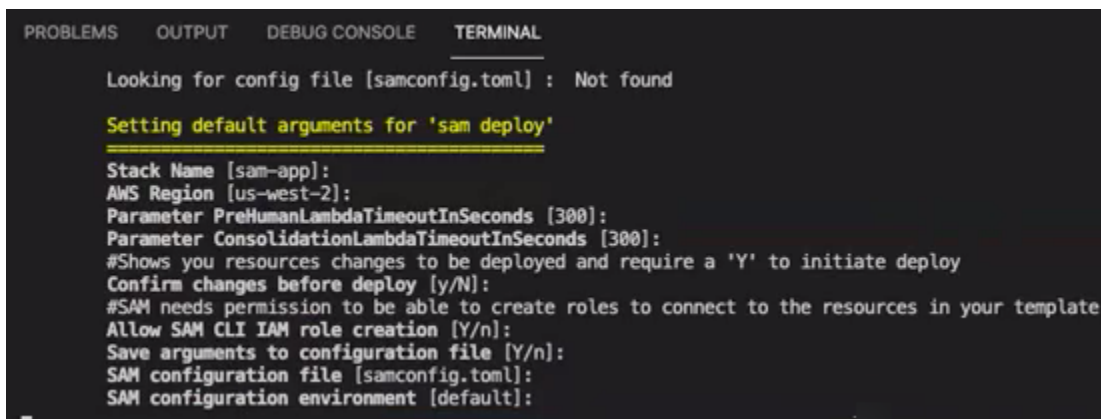
### Einrichten Ihrer Umgebung

1. Wenn Sie Windows verwenden, installieren Sie [Cygwin](#). Wenn Sie Linux oder Mac verwenden, überspringen Sie diesen Schritt.
2. Laden Sie die [Annotationsartefakte](#) von herunter. GitHub Entpacken Sie die Datei.
3. Navigieren Sie in Ihrem Terminalfenster zu dem entpackten Ordner (amazon-comprehend-semi-structured- documents-annotation-tools-main).
4. Dieser Ordner enthält eine Auswahl von OptionenMakefiles, die Sie ausführen, um Abhängigkeiten zu installieren, eine virtuelle Python-Umgebung einzurichten und die erforderlichen Ressourcen bereitzustellen. Lesen Sie die Readme-Datei, um Ihre Wahl zu treffen.

- Die empfohlene Option verwendet einen einzigen Befehl, um alle Abhängigkeiten in einer virtuellen Umgebung zu installieren, erstellt den AWS CloudFormation Stack aus der Vorlage und stellt den Stack mit interaktiver Anleitung für Sie bereit. AWS-Konto Führen Sie den folgenden Befehl aus:

```
make ready-and-deploy-guided
```

Dieser Befehl bietet eine Reihe von Konfigurationsoptionen. Vergewissern Sie sich, dass Sie AWS-Region korrekt sind. Für alle anderen Felder können Sie entweder die Standardwerte akzeptieren oder benutzerdefinierte Werte eingeben. Wenn Sie den AWS CloudFormation Stacknamen ändern, notieren Sie ihn in den nächsten Schritten so, wie Sie ihn benötigen.



```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
Looking for config file [samconfig.toml] : Not found

Setting default arguments for 'sam deploy'

Stack Name [sam-app]:
AWS Region [us-west-2]:
Parameter PreHumanLambdaTimeoutInSeconds [300]:
Parameter ConsolidationLambdaTimeoutInSeconds [300]:
#Shows you resources changes to be deployed and require a 'Y' to initiate deploy
Confirm changes before deploy [y/N]:
#SAM needs permission to be able to create roles to connect to the resources in your template
Allow SAM CLI IAM role creation [Y/n]:
Save arguments to configuration file [Y/n]:
SAM configuration file [samconfig.toml]:
SAM configuration environment [default]:

```

Der CloudFormation Stack erstellt und verwaltet die [AWS Lambdas](#), [AWS IAM-Rollen](#) und [AWS S3-Buckets](#), die für das [Annotationstool](#) erforderlich sind.

Sie können jede dieser Ressourcen auf der Seite mit den Stack-Details in der Konsole überprüfen. CloudFormation

- Der Befehl fordert Sie auf, die Bereitstellung zu starten. CloudFormation erstellt alle Ressourcen in der angegebenen Region.



```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
Deploying with following values
Stack name           : sam-app
Region               : us-west-2
Confirm changeset    : False
Deployment s3 bucket : aws-sam-cli-managed-default-samclisourcebucket-jrw8g1gm4pqh
Capabilities         : ["CAPABILITY_IAM"]
Parameter overrides  : {"PreHumanLambdaTimeoutInSeconds": "300", "ConsolidationLambdaTimeoutInSeconds": "300"}
Signing Profiles      : {}

Initiating deployment

```

Wenn der CloudFormation Stack-Status auf Create-Complete wechselt, sind die Ressourcen einsatzbereit.

## Eine PDF-Datei in einen S3-Bucket hochladen

Im Abschnitt [Einrichtung](#) haben Sie einen CloudFormation Stack bereitgestellt, der einen S3-Bucket mit dem Namen `comprehend-semi-structured-documents-${AWS::Region}-${AWS::AccountId}` erstellt. Sie laden jetzt Ihre PDF-Quelldokumente in diesen Bucket hoch.

### Note

Dieser Bucket enthält die Daten, die für Ihren Labeling-Job erforderlich sind. Die Richtlinie „Lambda Execution Role“ gewährt der Lambda-Funktion die Erlaubnis, auf diesen Bucket zuzugreifen.

Den Namen des S3-Buckets finden Sie in den CloudFormation Stack-Details mithilfe des Schlüssels „SemiStructuredDocumentsS3Bucket“.

1. Erstellen Sie einen neuen Ordner im S3-Bucket. Nennen Sie diesen neuen Ordner 'src'.
2. Fügen Sie Ihre PDF-Quelldateien zu Ihrem Ordner 'src' hinzu. In einem späteren Schritt fügen Sie diesen Dateien Anmerkungen hinzu, um Ihr Erkennungsprogramm zu trainieren.
3. (Optional) Hier ist ein AWS CLI-Beispiel, mit dem Sie Ihre Quelldokumente aus einem lokalen Verzeichnis in einen S3-Bucket hochladen können:

```
aws s3 cp --recursive local-path-to-your-source-docs s3://deploy-guided/src/
```

Oder mit Ihrer Region und Konto-ID:

```
aws s3 cp --recursive local-path-to-your-source-docs s3://deploy-guided-Region-AccountID/src/
```

4. Sie haben jetzt eine private SageMaker Ground-Truth-Belegschaft und haben Ihre Quelldateien in den S3-Bucket, `deploy-guided/src/`, hochgeladen. Jetzt können Sie mit dem Kommentieren beginnen.

## Einen Annotationsjob erstellen

Das Skript `comprehend-ssie-annotation-tool-cli.py` im `bin` Verzeichnis ist ein einfacher Wrapper-Befehl, der die Erstellung eines SageMaker Ground Truth-Labeling-Jobs vereinfacht. Das Python-Skript liest die Quelldokumente aus Ihrem S3-Bucket und erstellt eine entsprechende einseitige

Manifestdatei mit einem Quelldokument pro Zeile. Das Skript erstellt dann einen Labeling-Job, für den die Manifestdatei als Eingabe erforderlich ist.

Das Python-Skript verwendet den S3-Bucket und den CloudFormation S3-Stack, die Sie im Abschnitt [Einrichtung](#) konfiguriert haben. Zu den erforderlichen Eingabeparametern für das Skript gehören:

- `input-s3-path`: S3-Uri zu den Quelldokumenten, die Sie in Ihren S3-Bucket hochgeladen haben. Zum Beispiel: `s3://deploy-guided/src/`. Sie können diesem Pfad auch Ihre Region und Konto-ID hinzufügen. Zum Beispiel: `s3://deploy-guided-Region-AccountID/src/`.
- `cfn-name`: Der CloudFormation Stack-Name. Wenn Sie den Standardwert für den Stacknamen verwendet haben, lautet Ihr CFN-Name `sam-app`.
- `work-team-name`: Der Name der Belegschaft, den Sie beim Aufbau der privaten Belegschaft in SageMaker Ground Truth erstellt haben.
- `job-name-prefix`: Das Präfix für den SageMaker Ground Truth Labeling-Job. Beachten Sie, dass für dieses Feld eine Beschränkung von 29 Zeichen gilt. An diesen Wert wird ein Zeitstempel angehängt. Zum Beispiel: `my-job-name-20210902T232116`.
- `EntityTypes`: Die Entitäten, die Sie während Ihres Labeling-Jobs verwenden möchten, getrennt durch Kommas. Diese Liste muss alle Entitäten enthalten, die Sie in Ihrem Trainingsdatensatz mit Anmerkungen versehen möchten. Der Ground Truth Labeling-Job zeigt nur diese Entitäten an, damit Annotatoren Inhalte in den PDF-Dokumenten beschriften können.

Um zusätzliche Argumente anzuzeigen, die das Skript unterstützt, verwenden Sie die `-h` Option zum Anzeigen des Hilfeinhalts.

- Führen Sie das folgende Skript mit den Eingabeparametern aus, wie in der vorherigen Liste beschrieben.

```
python bin/comprehend-ssie-annotation-tool-cli.py \  
--input-s3-path s3://deploy-guided-Region-AccountID/src/ \  
--cfn-name sam-app \  
--work-team-name my-work-team-name \  
--region us-east-1 \  
--job-name-prefix my-job-name-20210902T232116 \  
--entity-types "EntityA, EntityB, EntityC" \  
--annotator-metadata "key=info,value=sample,key=Due Date,value=12/12/2021"
```

Das Skript erzeugt die folgende Ausgabe:

```
Downloaded files to temp local directory /tmp/a1dc0c47-0f8c-42eb-9033-74a988ccc5aa
Deleted downloaded temp files from /tmp/a1dc0c47-0f8c-42eb-9033-74a988ccc5aa
Uploaded input manifest file to s3://comprehend-semi-structured-documents-
us-west-2-123456789012/input-manifest/my-job-name-20220203-labeling-
job-20220203T183118.manifest
Uploaded schema file to s3://comprehend-semi-structured-documents-us-
west-2-123456789012/comprehend-semi-structured-docs-ui-template/my-job-
name-20220203-labeling-job-20220203T183118/ui-template/schema.json
Uploaded template UI to s3://comprehend-semi-structured-documents-us-
west-2-123456789012/comprehend-semi-structured-docs-ui-template/my-job-
name-20220203-labeling-job-20220203T183118/ui-template/template-2021-04-15.liquid
Sagemaker GroundTruth Labeling Job submitted: arn:aws:sagemaker:us-
west-2:123456789012:labeling-job/my-job-name-20220203-labeling-job-20220203t183118
(amazon-comprehend-semi-structured-documents-annotation-tools-main)
  user@3c063014d632 amazon-comprehend-semi-structured-documents-annotation-tools-
main %
```

## Kommentieren mit SageMaker Ground Truth

Nachdem Sie die erforderlichen Ressourcen konfiguriert und einen Labeling-Job erstellt haben, können Sie sich beim Labeling-Portal anmelden und Ihre PDF-Dateien mit Anmerkungen versehen.

1. Melden Sie sich mit den [SageMaker Webbrowsern Chrome oder Firefox bei der Konsole](#) an.
2. Wählen Sie Labeling Workforces und anschließend Private aus.
3. Wählen Sie unter Übersicht über private Mitarbeiter die Anmelde-URL für das Labeling-Portal aus, die Sie zusammen mit Ihren privaten Mitarbeitern erstellt haben. Melden Sie sich mit den entsprechenden Anmeldeinformationen an.

Wenn keine Jobs aufgelistet sind, machen Sie sich keine Sorgen. Die Aktualisierung kann je nach Anzahl der Dateien, die Sie zur Anmerkung hochgeladen haben, eine Weile dauern.

4. Wählen Sie Ihre Aufgabe aus und klicken Sie in der oberen rechten Ecke auf Mit der Arbeit beginnen, um den Kommentarbildschirm zu öffnen.

Auf dem Kommentarbildschirm wird eines Ihrer Dokumente geöffnet und darüber werden die Entitätstypen angezeigt, die Sie bei der Einrichtung angegeben haben. Rechts neben Ihren Entitätstypen befindet sich ein Pfeil, mit dem Sie durch Ihre Dokumente navigieren können.

Instructions Shortcuts

Labeling Task: **NER** **OFFERING\_PRICE** **OFFERED\_SHARES** << 2 >>

[Table of Contents](#)

### DILUTION

If you purchase units in this offering, you will experience dilution to the extent of the difference between the public offering price of the units (attributing no value to the warrants) and the net tangible book value per share of our common stock immediately after this offering.

Our net tangible book value as of June 30, 2017 was approximately \$15.9 million, or \$0.5589 per share of common stock. Net tangible book value per share is equal to our total tangible assets minus total liabilities, all divided by the number of shares of common stock outstanding as of June 30, 2017.

After giving effect to the sale of **OFFERED\_SHARES** units at a price of **OFFERING\_PRICE** per unit, and after deducting our estimated placement agent fees and offering expenses payable by us, and attributing no value to the warrants, our as adjusted net tangible book value would have been approximately \$23.3 million, or approximately \$0.7357 per share of common stock, as of June 30, 2017. This represents an immediate increase in net tangible book value of approximately \$0.1768 per share to existing stockholders and an immediate dilution of approximately \$1.714 per share to new investors. The following table illustrates this calculation on a per share basis:

Public offering price per unit		<b>OFFERING_PRICE</b> \$ 2.45
Net tangible book value per share as of June 30, 2017	\$ 0.5589	
Increase per share attributable to this offering	\$ 0.1768	
As adjusted net tangible book value per share as of June 30, 2017, after giving effect to this offering		\$ 0.7357
Dilution per share to new investors		\$ 1.714

The foregoing table and discussion is based on 28,452,305 shares outstanding as of June 30, 2017 and excludes:

- 1,937,871 shares of our common stock subject to outstanding options having a weighted average exercise price of \$5.54 per share;
- 54,300 shares of our common stock subject to outstanding restricted stock units;

Kommentieren Sie das geöffnete Dokument. Sie können Ihre Anmerkungen in jedem Dokument auch entfernen, rückgängig machen oder auto taggen. Diese Optionen sind im rechten Bereich des Kommentarwerkzeugs verfügbar.

**METADATA**

**BLOCK** **SHOW DOCUMENT**

**SELECTED ENTITIES** COPY REMOVE

UNASSIGNED Repeat Location.

0 16

RESET

**ENTITY LIST 1** REMOVE ALL

Um Auto-Tag zu verwenden, kommentieren Sie eine Instanz einer Ihrer Entitäten. Alle anderen Vorkommen dieses bestimmten Wortes werden dann automatisch mit diesem Entitätstyp annotiert.

Wenn Sie fertig sind, wählen Sie unten rechts Senden aus und verwenden Sie dann die Navigationspfeile, um zum nächsten Dokument zu gelangen. Wiederholen Sie diesen Vorgang, bis Sie alle Ihre PDF-Dateien mit Anmerkungen versehen haben.

Nachdem Sie alle Schulungsdokumente mit Anmerkungen versehen haben, finden Sie die Anmerkungen im JSON-Format im Amazon S3 S3-Bucket an dieser Stelle:

```
/output/your labeling job name/annotations/
```

Der Ausgabeordner enthält auch eine Ausgabemanifestdatei, in der alle Anmerkungen in Ihren Schulungsdokumenten aufgeführt sind. Sie finden Ihre Ausgabe-Manifestdatei am folgenden Speicherort.

```
/output/your labeling job name/manifests/
```

## Trainieren von benutzerdefinierten Entity Recognizer-Modellen

Ein benutzerdefinierter Entity Recognizer identifiziert nur die Entitätstypen, die Sie beim Trainieren des Modells einbeziehen. Es enthält nicht automatisch die voreingestellten Entitätstypen. Wenn Sie auch die voreingestellten Entitätstypen wie LOCATION, DATE oder Bol identifizieren möchten, müssen Sie zusätzliche Trainingsdaten für diese Entitäten bereitstellen.

Wenn Sie einen benutzerdefinierten Entity Recognizer mithilfe von kommentierten PDF-Dateien erstellen, können Sie den Recognizer mit einer Vielzahl von Eingabedateiformaten verwenden: Klartext, Bilddateien (JPG, PNG, TIFF), PDF-Dateien und Word-Dokumente, ohne dass eine Vorverarbeitung oder eine Dokumentabflachung erforderlich ist. Amazon Comprehend unterstützt keine Anmerkung von Bilddateien oder Word-Dokumenten.

### Note

Ein benutzerdefinierter Entity Recognizer, der kommentierte PDF-Dateien verwendet, unterstützt nur englische Dokumente.

Nachdem Sie einen benutzerdefinierten Entity Recognizer erstellt haben, können Sie den Fortschritt der Anforderung mithilfe der [-DescribeEntityRecognizer](#) Operation überwachen. Sobald das Status Feld ist TRAINED, ist das Erkennungsmodell bereit für die Erkennung benutzerdefinierter Entitäten.

## Themen

- [Trainieren von benutzerdefinierten Erkennern \(Konsole\)](#)
- [Trainieren von benutzerdefinierten Entity Recognizern \(API\)](#)
- [Benutzerdefinierte Metriken zur Erkennung von Entitäten](#)

## Trainieren von benutzerdefinierten Erkennern (Konsole)

Mit der Amazon Comprehend Comprehend-Konsole können Sie benutzerdefinierte Entitätserkennung erstellen. In diesem Abschnitt erfahren Sie, wie Sie einen benutzerdefinierten Entitätserkennung erstellen und trainieren.

Erstellen eines benutzerdefinierten Entitätserkenners mithilfe der Konsole im CSV-Format

Um den benutzerdefinierten Entitätserkennung zu erstellen, stellen Sie zunächst einen Datensatz bereit, mit dem Sie Ihr Modell trainieren können. Fügen Sie diesem Datensatz eines der folgenden Elemente bei: eine Reihe von Dokumenten mit Anmerkungen oder eine Liste von Entitäten und deren Typbezeichnung sowie eine Reihe von Dokumenten, die diese Entitäten enthalten. Weitere Informationen finden Sie unter [Erkennung benutzerdefinierter Entitäten](#).


Um einen benutzerdefinierten Entitätserkennung mit einer CSV-Datei zu trainieren

1. [Melden Sie sich bei der Amazon Comprehend Comprehend-Konsole an AWS Management Console und öffnen Sie sie unter <https://console.aws.amazon.com/comprehend/>](#)
2. Wählen Sie im linken Menü Anpassung und dann Benutzerdefinierte Entitätserkennung aus.
3. Wählen Sie Neues Modell erstellen aus.
4. Geben Sie dem Recognizer einen Namen. Der Name muss innerhalb der Region und des Kontos eindeutig sein.
5. Wählen Sie die Sprache aus.
6. Geben Sie unter Benutzerdefinierter Entitätstyp eine benutzerdefinierte Bezeichnung ein, die der Recognizer im Datensatz finden soll.

Der Entitätstyp muss in Großbuchstaben geschrieben werden. Wenn er aus mehr als einem Wort besteht, trennen Sie die Wörter durch einen Unterstrich.



7. Wählen Sie Typ hinzufügen.
8. Wenn Sie einen zusätzlichen Entitätstyp hinzufügen möchten, geben Sie ihn ein und wählen Sie dann Typ hinzufügen. Wenn Sie einen der hinzugefügten Entitätstypen entfernen möchten, wählen Sie Typ entfernen und wählen Sie dann den Entitätstyp aus, der aus der Liste entfernt werden soll. Es können maximal 25 Entitätstypen aufgelistet werden.
9. Um Ihren Schulungsjob zu verschlüsseln, wählen Sie Recognizer-Verschlüsselung und dann aus, ob Sie einen KMS-Schlüssel verwenden möchten, der mit dem aktuellen Konto verknüpft ist, oder einen von einem anderen Konto.
  - Wenn Sie einen Schlüssel verwenden, der dem aktuellen Konto zugeordnet ist, wählen Sie als KMS-Schlüssel-ID die Schlüssel-ID aus.
  - Wenn Sie einen Schlüssel verwenden, der einem anderen Konto zugeordnet ist, geben Sie für KMS-Schlüssel-ARN den ARN für die Schlüssel-ID ein.

 Note

Weitere Informationen zur Erstellung und Verwendung von KMS-Schlüsseln und der zugehörigen Verschlüsselung finden Sie unter [AWS Key Management Service](#).

10. Wählen Sie unter Datenspezifikationen das Format Ihrer Schulungsdokumente aus:
  - CSV-Datei — Eine CSV-Datei, die Ihre Schulungsdokumente ergänzt. Die CSV-Datei enthält Informationen über die benutzerdefinierten Entitäten, die Ihr trainiertes Modell erkennen wird. Das erforderliche Format der Datei hängt davon ab, ob Sie Anmerkungen oder eine Entitätsliste bereitstellen.
  - Erweitertes Manifest — Ein beschrifteter Datensatz, der von Amazon SageMaker Ground Truth erstellt wurde. Diese Datei ist im JSON-Zeilenformat. Jede Zeile ist ein vollständiges JSON-Objekt, das ein Schulungsdokument und dessen Beschriftungen enthält. Jedes Label kennzeichnet eine benannte Entität im Schulungsdokument. Sie können bis zu 5 erweiterte Manifestdateien bereitstellen.

Weitere Informationen zu verfügbaren Formaten und Beispielen finden Sie unter [Trainieren von benutzerdefinierten Entity Recognizer-Modellen](#).

11. Wählen Sie unter Trainingstyp den zu verwendenden Trainingstyp aus:
  - Verwenden von Anmerkungen und Schulungsdokumenten

- Verwenden von Entitätslisten und Schulungsdokumenten

Wenn Sie Anmerkungen auswählen, geben Sie die URL der Annotationsdatei in Amazon S3 ein. Sie können auch zu dem Bucket oder Ordner in Amazon S3 navigieren, in dem sich die Annotationsdateien befinden, und S3 durchsuchen wählen.

Wenn Sie Entitätsliste wählen, geben Sie die URL der Entitätsliste in Amazon S3 ein. Sie können auch zu dem Bucket oder Ordner in Amazon S3 navigieren, in dem sich die Entitätsliste befindet, und S3 durchsuchen auswählen.

12. Geben Sie die URL eines Eingabedatensatzes ein, der die Schulungsdokumente in Amazon S3 enthält. Sie können auch zu dem Bucket oder Ordner in Amazon S3 navigieren, in dem sich die Schulungsdokumente befinden, und Ordner auswählen wählen.
13. Wählen Sie unter Testdatensatz aus, wie Sie die Leistung Ihres trainierten Modells bewerten möchten. Sie können dies sowohl für Annotationen als auch für Trainingstypen mit Entitätslisten tun.
  - Autosplit: Autosplit wählt automatisch 10% der von Ihnen bereitgestellten Trainingsdaten aus, um sie als Testdaten zu verwenden
  - (Optional) Vom Kunden bereitgestellt: Wenn Sie vom Kunden bereitgestellt auswählen, können Sie genau angeben, welche Testdaten Sie verwenden möchten.
14. Wenn Sie vom Kunden bereitgestellter Testdatensatz auswählen, geben Sie die URL der Annotationsdatei in Amazon S3 ein. Sie können auch zu dem Bucket oder Ordner in Amazon S3 navigieren, in dem sich die Annotationsdateien befinden, und Ordner auswählen wählen.
15. Wählen Sie im Abschnitt Wählen Sie eine IAM-Rolle entweder eine bestehende IAM-Rolle aus oder erstellen Sie eine neue.
  - Wählen Sie eine bestehende IAM-Rolle — Wählen Sie diese Option, wenn Sie bereits über eine IAM-Rolle mit Zugriffsberechtigungen für die Eingabe- und Ausgabe-Buckets von Amazon S3 verfügen.
  - Neue IAM-Rolle erstellen — Wählen Sie diese Option, wenn Sie eine neue IAM-Rolle mit den entsprechenden Berechtigungen für Amazon Comprehend für den Zugriff auf die Eingabe- und Ausgabe-Buckets erstellen möchten.

**Note**

Wenn die Eingabedokumente verschlüsselt sind, muss die verwendete IAM-Rolle über eine entsprechende Berechtigung verfügen. `kms:Decrypt` Weitere Informationen finden Sie unter [Erforderliche Berechtigungen für die Verwendung der KMS-Verschlüsselung](#).

16. (Optional) Um Ihre Ressourcen von einer VPC aus in Amazon Comprehend zu starten, geben Sie die VPC-ID unter VPC ein oder wählen Sie die ID aus der Drop-down-Liste aus.
  1. Wählen Sie das Subnetz unter Subnetz (s) aus. Nachdem Sie das erste Subnetz ausgewählt haben, können Sie weitere auswählen.
  2. Wählen Sie unter Sicherheitsgruppe (n) die zu verwendende Sicherheitsgruppe aus, falls Sie eine angegeben haben. Nachdem Sie die erste Sicherheitsgruppe ausgewählt haben, können Sie weitere auswählen.

**Note**

Wenn Sie eine VPC mit Ihrem benutzerdefinierten Entitätserkennungsjob verwenden, müssen die für die Vorgänge `Create` und `StartDataAccessRole` verwendeten über Berechtigungen für die VPC verfügen, von der aus auf die Eingabedokumente und den Ausgabe-Bucket zugegriffen wird.

17. (Optional) Um dem benutzerdefinierten Entity Recognizer ein Tag hinzuzufügen, geben Sie unter Tags ein Schlüssel-Wert-Paar ein. Wählen Sie `Add tag`. Um dieses Paar zu entfernen, bevor Sie den Recognizer erstellen, wählen Sie `Tag entfernen` aus.
18. Wählen Sie `Zug`.

Der neue Recognizer erscheint dann in der Liste und zeigt seinen Status an. Er wird zuerst als `Submitted` angezeigt. Es wird dann `Training` für einen Klassifikator angezeigt, der Schulungsdokumente verarbeitet, `Trained` für einen Klassifikator, der sofort verwendet werden kann, und `In error` für einen Klassifikator, der einen Fehler aufweist. Sie können auf einen Job klicken, um weitere Informationen über den Recognizer zu erhalten, einschließlich aller Fehlermeldungen.


## Erstellen eines benutzerdefinierten Entitätserkenners mithilfe der Konsole — erweitertes Manifest

Um einen benutzerdefinierten Entitätserkennung mit einem Klartext-, PDF- oder Word-Dokument zu trainieren

1. Melden Sie sich bei der [Amazon Comprehend Comprehend-Konsole](#) an AWS Management Console und öffnen Sie sie.
2. Wählen Sie im linken Menü Anpassung und dann Benutzerdefinierte Entitätserkennung aus.
3. Wählen Sie Train Recognizer.
4. Geben Sie dem Recognizer einen Namen. Der Name muss innerhalb der Region und des Kontos eindeutig sein.
5. Wählen Sie die Sprache aus. Hinweis: Wenn Sie ein PDF- oder Word-Dokument trainieren, ist Englisch die unterstützte Sprache.
6. Geben Sie unter Benutzerdefinierter Entitätstyp eine benutzerdefinierte Bezeichnung ein, die der Recognizer im Datensatz finden soll.

Der Entitätstyp muss in Großbuchstaben geschrieben werden. Wenn er aus mehr als einem Wort besteht, trennen Sie die Wörter durch einen Unterstrich.

7. Wählen Sie Typ hinzufügen.
8. Wenn Sie einen zusätzlichen Entitätstyp hinzufügen möchten, geben Sie ihn ein und wählen Sie dann Typ hinzufügen. Wenn Sie einen der hinzugefügten Entitätstypen entfernen möchten, wählen Sie Typ entfernen und wählen Sie dann den Entitätstyp aus, der aus der Liste entfernt werden soll. Es können maximal 25 Entitätstypen aufgelistet werden.
9. Um Ihren Schulungsjob zu verschlüsseln, wählen Sie Recognizer-Verschlüsselung und dann aus, ob Sie einen KMS-Schlüssel verwenden möchten, der mit dem aktuellen Konto verknüpft ist, oder einen von einem anderen Konto.
  - Wenn Sie einen Schlüssel verwenden, der dem aktuellen Konto zugeordnet ist, wählen Sie als KMS-Schlüssel-ID die Schlüssel-ID aus.
  - Wenn Sie einen Schlüssel verwenden, der einem anderen Konto zugeordnet ist, geben Sie für KMS-Schlüssel-ARN den ARN für die Schlüssel-ID ein.

 Note

Weitere Informationen zur Erstellung und Verwendung von KMS-Schlüsseln und der zugehörigen Verschlüsselung finden Sie unter [AWS Key Management Service](#).

10. Wählen Sie unter Trainingsdaten Augmented Manifest als Datenformat aus:

- Augmented Manifest — ist ein beschrifteter Datensatz, der von Amazon SageMaker Ground Truth erstellt wird. Diese Datei ist im JSON-Zeilenformat. Jede Zeile in der Datei ist ein vollständiges JSON-Objekt, das ein Schulungsdokument und dessen Beschriftungen enthält. Jedes Label kennzeichnet eine benannte Entität im Schulungsdokument. Sie können bis zu 5 erweiterte Manifestdateien bereitstellen. Wenn Sie PDF-Dokumente für Trainingsdaten verwenden, müssen Sie Augmented Manifest auswählen. Sie können bis zu 5 erweiterte Manifestdateien bereitstellen. Für jede Datei können Sie bis zu 5 Attribute benennen, die als Trainingsdaten verwendet werden sollen.


Weitere Informationen zu verfügbaren Formaten und Beispielen finden Sie unter [Trainieren von benutzerdefinierten Entity Recognizer-Modellen](#).

11. Wählen Sie den Trainingsmodelltyp aus.

Wenn Sie Klartext-Dokumente ausgewählt haben, geben Sie unter Eingabespeicherort die Amazon S3-URL der erweiterten Manifestdatei von Amazon SageMakerGround Truth ein. Sie können auch zu dem Bucket oder Ordner in Amazon S3 navigieren, in dem sich die erweiterten Manifeste befinden, und Ordner auswählen auswählen.


12. Geben Sie unter Attributname den Namen des Attributs ein, das Ihre Anmerkungen enthält. Wenn die Datei Anmerkungen aus mehreren verketteten Labeling-Jobs enthält, fügen Sie für jeden Job ein Attribut hinzu. In diesem Fall enthält jedes Attribut den Satz von Anmerkungen aus einem Labeling-Job. Hinweis: Sie können bis zu 5 Attributnamen für jede Datei angeben.
13. Wählen Sie Hinzufügen aus.
14. Wenn Sie unter Eingabespeicherort die Option PDF, Word-Dokumente ausgewählt haben, geben Sie die Amazon S3-URL der erweiterten Manifestdatei von Amazon SageMaker Ground Truth ein. Sie können auch zu dem Bucket oder Ordner in Amazon S3 navigieren, in dem sich die erweiterten Manifeste befinden, und Ordner auswählen auswählen.
15. Geben Sie das S3-Präfix für Ihre Annotation-Datendateien ein. Dies sind die PDF-Dokumente, die Sie beschriftet haben.

16. Geben Sie das S3-Präfix für Ihre Quelldokumente ein. Dies sind die Original-PDF-Dokumente (Datenobjekte), die Sie Ground Truth für Ihre Etikettierungsaufgabe zur Verfügung gestellt haben.
17. Geben Sie die Attributnamen ein, die Ihre Anmerkungen enthalten. Hinweis: Sie können bis zu 5 Attributnamen für jede Datei angeben. Alle Attribute in Ihrer Datei, die Sie nicht angeben, werden ignoriert.
18. Wählen Sie im Abschnitt IAM-Rolle entweder eine vorhandene IAM-Rolle aus oder erstellen Sie eine neue.
  - Wählen Sie eine bestehende IAM-Rolle — Wählen Sie diese Option, wenn Sie bereits über eine IAM-Rolle mit Zugriffsberechtigungen für die Eingabe- und Ausgabe-Buckets von Amazon S3 verfügen.
  - Neue IAM-Rolle erstellen — Wählen Sie diese Option, wenn Sie eine neue IAM-Rolle mit den entsprechenden Berechtigungen für Amazon Comprehend für den Zugriff auf die Eingabe- und Ausgabe-Buckets erstellen möchten.

 Note

Wenn die Eingabedokumente verschlüsselt sind, muss die verwendete IAM-Rolle über eine entsprechende Berechtigung verfügen. `kms:Decrypt` Weitere Informationen finden Sie unter [Erforderliche Berechtigungen für die Verwendung der KMS-Verschlüsselung](#).

19. (Optional) Um Ihre Ressourcen von einer VPC aus in Amazon Comprehend zu starten, geben Sie die VPC-ID unter VPC ein oder wählen Sie die ID aus der Drop-down-Liste aus.
  1. Wählen Sie das Subnetz unter Subnetz (s) aus. Nachdem Sie das erste Subnetz ausgewählt haben, können Sie weitere auswählen.
  2. Wählen Sie unter Sicherheitsgruppe (n) die zu verwendende Sicherheitsgruppe aus, falls Sie eine angegeben haben. Nachdem Sie die erste Sicherheitsgruppe ausgewählt haben, können Sie weitere auswählen.

 Note

Wenn Sie eine VPC mit Ihrem benutzerdefinierten Entitätserkennungsjob verwenden, müssen die für die Vorgänge `Create` und `StartDataAccessRole` verwendeten über

Berechtigungen für die VPC verfügen, von der aus auf die Eingabedokumente und den Ausgabe-Bucket zugegriffen wird.

20. (Optional) Um dem benutzerdefinierten Entity Recognizer ein Tag hinzuzufügen, geben Sie unter Tags ein Schlüssel-Wert-Paar ein. Wählen Sie Add tag. Um dieses Paar zu entfernen, bevor Sie den Recognizer erstellen, wählen Sie Tag entfernen aus.
21. Wählen Sie Zug.

Der neue Recognizer erscheint dann in der Liste und zeigt seinen Status an. Er wird zuerst als Submitted angezeigt. Es wird dann Training für einen Klassifikator angezeigt, der Schulungsdokumente verarbeitet, Trained für einen Klassifikator, der sofort verwendet werden kann, und In error für einen Klassifikator, der einen Fehler aufweist. Sie können auf einen Job klicken, um weitere Informationen über den Recognizer zu erhalten, einschließlich aller Fehlermeldungen.

## Trainieren von benutzerdefinierten Entity Recognizern (API)

Um ein benutzerdefiniertes Entitätserkennungsmodell zu erstellen und zu trainieren, verwenden Sie die Amazon Comprehend [CreateEntityRecognizer](#) API-Operation

### Themen

- [Trainieren von benutzerdefinierten Entitätserkennungen mithilfe der AWS Command Line Interface](#)
- [Trainieren von benutzerdefinierten Entitätserkennungen mithilfe der AWS SDK for Java](#)
- [Trainieren von benutzerdefinierten Entitätserkennungen mit Python \(Boto3\)](#)

## Trainieren von benutzerdefinierten Entitätserkennungen mithilfe der AWS Command Line Interface

Die folgenden Beispiele zeigen die Verwendung der `-CreateEntityRecognizerOperation` und anderer zugehöriger APIs mit der AWS CLI.

Die Beispiele sind für Unix, Linux und macOS formatiert. Ersetzen Sie unter Windows den umgekehrten Schrägstrich (`\`), das Unix-Fortsetzungszeichen, am Ende jeder Zeile durch ein Caret-Zeichen oder Zirkumflex (`^`).

Erstellen Sie einen benutzerdefinierten Entity Recognizer mit dem `create-entity-recognizer` CLI-Befehl . Weitere Informationen zum `input-data-config` Parameter finden Sie unter [CreateEntityRecognizer](#) in der Amazon Comprehend API-Referenz.

```
aws comprehend create-entity-recognizer \  
  --language-code en \  
  --recognizer-name test-6 \  
  --data-access-role-arn "arn:aws:iam::account number:role/service-role/  
AmazonComprehendServiceRole-role" \  
  --input-data-config "EntityTypes=[{Type=PERSON}], Documents={S3Uri=s3://Bucket  
Name/Bucket Path/documents},  
                        Annotations={S3Uri=s3://Bucket Name/Bucket Path/annotations}" \  
  --region region
```

Listen Sie alle Entity Recognizer in einer Region mit dem `list-entity-recognizers` CLI-Befehl auf.

```
aws comprehend list-entity-recognizers \  
  --region region
```

Überprüfen Sie den Auftragsstatus von benutzerdefinierten Entitätserkennungen mit dem `describe-entity-recognizer` CLI-Befehl .

```
aws comprehend describe-entity-recognizer \  
  --entity-recognizer-arn arn:aws:comprehend:region:account number:entity-  
recognizer/test-6 \  
  --region region
```

## Trainieren von benutzerdefinierten Entitätserkennungen mithilfe der AWS SDK for Java

In diesem Beispiel wird ein benutzerdefinierter Entity Recognizer erstellt und das Modell mithilfe von Java trainiert.

Beispiele für Amazon Comprehend, die Java verwenden, finden Sie unter [Amazon Comprehend Java-Beispiele](#).

## Trainieren von benutzerdefinierten Entitätserkennungen mit Python (Boto3)

Boto3 SDK instanziiieren:



```
import boto3
import uuid
comprehend = boto3.client("comprehend", region_name="region")
```

Entitätserkennung erstellen:

```
response = comprehend.create_entity_recognizer(
    RecognizerName="Recognizer-Name-Goes-Here-{}".format(str(uuid.uuid4()))),
    LanguageCode="en",
    DataAccessRoleArn="Role ARN",
    InputDataConfig={
        "EntityTypes": [
            {
                "Type": "ENTITY_TYPE"
            }
        ],
        "Documents": {
            "S3Uri": "s3://Bucket Name/Bucket Path/documents"
        },
        "Annotations": {
            "S3Uri": "s3://Bucket Name/Bucket Path/annotations"
        }
    }
)
recognizer_arn = response["EntityRecognizerArn"]
```

Listet alle Erkennungen auf:

```
response = comprehend.list_entity_recognizers()
```

Warten Sie, bis die Erkennung den Status TRAINED erreicht:

```
while True:
    response = comprehend.describe_entity_recognizer(
        EntityRecognizerArn=recognizer_arn
    )

    status = response["EntityRecognizerProperties"]["Status"]
    if "IN_ERROR" == status:
        sys.exit(1)
    if "TRAINED" == status:
```

```
break
```

```
time.sleep(10)
```

## Benutzerdefinierte Metriken zur Erkennung von Entitäten

Amazon Comprehend stellt Ihnen Kennzahlen zur Verfügung, anhand derer Sie abschätzen können, wie gut ein Entity Recognizer für Ihre Aufgabe funktionieren sollte. Sie basieren auf dem Training des Erkennungsmodells. Sie stellen also zwar die Leistung des Modells während des Trainings genau dar, stellen jedoch nur eine Annäherung an die API-Leistung bei der Entitätserkennung dar.

Metriken werden jedes Mal zurückgegeben, wenn Metadaten eines trainierten Entitätserkenners zurückgegeben werden.

Amazon Comprehend unterstützt das Trainieren eines Modells für bis zu 25 Entitäten gleichzeitig. Wenn Metriken von einem trainierten Entity-Recognizer zurückgegeben werden, werden die Ergebnisse sowohl für den Recognizer als Ganzes (globale Metriken) als auch für jede einzelne Entität (Entitätsmetriken) berechnet.

Drei Metriken sind verfügbar, sowohl als globale als auch als Entitätsmetriken:

- Präzision

Dies gibt den Anteil der vom System erstellten Entitäten an, die korrekt identifiziert und beschriftet wurden. Dies zeigt, wie oft die Entitätsidentifikation des Modells eine wirklich gute Identifikation ist. Dies ist ein Prozentsatz der Gesamtzahl der Identifizierungen.

Mit anderen Worten, die Genauigkeit basiert auf wahren positiven (tp) und falsch positiven (fp) und wird als  $\text{Genauigkeit} = \text{tp} / (\text{tp} + \text{fp})$  berechnet.

Wenn ein Modell beispielsweise voraussagt, dass zwei Beispiele für eine Entität in einem Dokument vorhanden sind, obwohl es eigentlich nur eines gibt, ist das Ergebnis ein richtig positives und ein falsch positives. In diesem Fall ist  $\text{Genauigkeit} = 1 / (1 + 1)$ . Die Genauigkeit beträgt 50%, da eine der beiden durch das Modell identifizierten Entitäten korrekt ist.


- Erinnern

Dies gibt den Anteil der in den Dokumenten vorhandenen Entitäten an, die vom System korrekt identifiziert und gekennzeichnet wurden. Mathematisch wird dies als die Gesamtzahl der korrekten Identifizierungen definiert: wahre positive (tp) und verpasste Identifizierungen falsch negative (fn).

Sie wird als  $\text{Recall} = \text{tp}/(\text{tp} + \text{fn})$  berechnet. Wenn ein Modell beispielsweise eine Entität korrekt identifiziert, aber zwei andere Instanzen übersieht, in denen diese Entität vorhanden ist, ist das Ergebnis eine echte positive und zwei falsch negative. In diesem Fall ist  $\text{recall} = 1/(1 + 2)$ . Der Rückruf liegt bei 33,33%, da eine von drei möglichen Beispielen richtig ist.

- F1-Ergebnis

Dies ist eine Kombination aus den Kennzahlen Precision und Recall, mit denen die Gesamtgenauigkeit des Modells für die benutzerdefinierte Entitätserkennung gemessen wird. Der F1-Score ist das harmonische Mittel der Messwerte Precision und Recall:  $F1 = 2 * \text{Precision} * \text{Recall}/(\text{Precision} + \text{Recall})$ .

 Note

Intuitiv gesehen benachteiligt das harmonische Mittel die Extreme stärker als der einfache Durchschnitt oder andere Mittelwerte (Beispiel:  $\text{precision} = 0$ ,  $\text{recall} = 1$ ) könnten trivial erreicht werden, indem alle möglichen Spannen vorhergesagt werden. Hier wäre der einfache Durchschnitt 0,5, würde ihn aber mit 0 bestrafen). F1

In den obigen Beispielen  $\text{precision} = 50\%$  und  $\text{recall} = 33,33\%$ , also  $F1 = 2 * 0,5 * 0,3333/(0,5 + 0,3333)$ . Der F1-Score ist 0,3975 oder 39,75%.

## Globale und individuelle Kennzahlen für Entitäten

Die Beziehung zwischen globalen und individuellen Entitätskennzahlen wird deutlich, wenn man den folgenden Satz für Entitäten analysiert, bei denen es sich entweder um einen Ort oder eine Person handelt

John Washington and his friend Smith live in San Francisco, work in San Diego, and own a house in Seattle.

In unserem Beispiel macht das Modell die folgenden Vorhersagen.

John Washington = Person  
Smith = Place  
San Francisco = Place

```
San Diego = Place
Seattle = Person
```

Die Vorhersagen hätten jedoch die folgenden sein müssen.

```
John Washington = Person
Smith = Person
San Francisco = Place
San Diego = Place
Seattle = Place
```

Die einzelnen Entitätskennzahlen hierfür wären:

entity: Person

True positive (TP) = 1 (because John Washington is correctly predicted to be a Person).

False positive (FP) = 1 (because Seattle is incorrectly predicted to be a Person, but is actually a Place).

False negative (FN) = 1 (because Smith is incorrectly predicted to be a Place, but is actually a Person).

Precision =  $1 / (1 + 1) = 0.5$  or 50%

Recall =  $1 / (1+1) = 0.5$  or 50%

F1 Score =  $2 * 0.5 * 0.5 / (0.5 + 0.5) = 0.5$  or 50%

entity: Place

TP = 2 (because San Francisco and San Diego are each correctly predicted to be a Place).

FP = 1 (because Smith is incorrectly predicted to be a Place, but is actually a Person).

FN = 1 (because Seattle is incorrectly predicted to be a Person, but is actually a Place).

Precision =  $2 / (2+1) = 0.6667$  or 66.67%

Recall =  $2 / (2+1) = 0.6667$  or 66.67%

F1 Score =  $2 * 0.6667 * 0.6667 / (0.6667 + 0.6667) = 0.6667$  or 66.67%

Die globalen Kennzahlen hierfür wären:

Weltweit:

Global:

TP = 3 (because John Washington, San Francisco and San Diego are predicted correctly).

```
This is also the sum of all individual entity TP).
FP = 2 (because Seattle is predicted as Person and Smith is predicted as Place. This
is the sum of all individual entity FP).
FN = 2 (because Seattle is predicted as Person and Smith is predicted as Place. This
is the sum of all individual FN).
Global Precision = 3 / (3+2) = 0.6 or 60%
(Global Precision = Global TP / (Global TP + Global FP))
Global Recall = 3 / (3+2) = 0.6 or 60%
(Global Recall = Global TP / (Global TP + Global FN))
Global F1Score = 2 * 0.6 * 0.6 / (0.6 + 0.6) = 0.6 or 60%
(Global F1Score = 2 * Global Precision * Global Recall / (Global Precision +
Global Recall))
```

## Verbesserung der Leistung des benutzerdefinierten Entity Recognizers

Diese Metriken geben Aufschluss darüber, wie genau das trainierte Modell funktioniert, wenn Sie es zur Identifizierung von Entitäten verwenden. Hier sind einige Optionen, mit denen Sie Ihre Metriken verbessern können, falls sie unter Ihren Erwartungen liegen:

1. Je nachdem, ob Sie [Anmerkungen](#) oder verwenden [Entitätslisten \(nur Klartext\)](#), achten Sie darauf, die Richtlinien in der jeweiligen Dokumentation zu befolgen, um die Datenqualität zu verbessern. Wenn Sie nach der Verbesserung Ihrer Daten und dem erneuten Training des Modells bessere Messwerte beobachten, können Sie die Datenqualität weiter verbessern, um eine bessere Modelleistung zu erzielen.
2. Wenn Sie eine Entitätsliste verwenden, sollten Sie stattdessen Anmerkungen verwenden. Manuelle Anmerkungen können Ihre Ergebnisse oft verbessern.
3. Wenn Sie sicher sind, dass kein Datenqualitätsproblem vorliegt und die Messwerte dennoch unverhältnismäßig niedrig sind, reichen Sie bitte eine Support-Anfrage ein.

## Ausführen einer benutzerdefinierten Erkennungsanalyse in Echtzeit

Echtzeitanalysen sind nützlich für Anwendungen, die kleine Dokumente verarbeiten, sobald sie eintreffen. Sie können beispielsweise benutzerdefinierte Entitäten in Social-Media-Beiträgen, Support-Tickets oder Kundenrezensionen erkennen.

### Bevor Sie beginnen

Sie benötigen ein benutzerdefiniertes Entitätserkennungsmodell (auch bekannt als Recognizer), bevor Sie benutzerdefinierte Entitäten erkennen können. Weitere Informationen zu diesen Modellen finden Sie unter [the section called “Trainingserkennungsmodelle”](#).

Ein Recognizer, der mit Klartext-Anmerkungen trainiert wurde, unterstützt nur die Entitätserkennung für Klartextdokumente. Ein Recognizer, der mit Anmerkungen zu PDF-Dokumenten trainiert wurde, unterstützt die Entitätserkennung für Klartextdokumente, Bilder, PDF-Dateien und Word-Dokumente. Informationen zu den Eingabedateien finden Sie unter [Eingaben für benutzerdefinierte Echtzeitanalysen](#).

Wenn Sie vorhaben, Bilddateien oder gescannte PDF-Dokumente zu analysieren, muss Ihre IAM-Richtlinie Berechtigungen zur Verwendung von zwei Amazon Textract-API-Methoden (DetectDocumentText und AnalyzeDocument) erteilen. Amazon Comprehend ruft diese Methoden während der Textextraktion auf. Eine Beispielrichtlinie finden Sie unter [Erforderliche Berechtigungen zum Ausführen von Dokumentanalyseaktionen](#).

#### Themen

- [Echtzeitanalyse für die Erkennung benutzerdefinierter Entitäten \(Konsole\)](#)
- [Echtzeitanalyse für die Erkennung benutzerdefinierter Entitäten \(API\)](#)
- [Ausgaben für Echtzeitanalysen](#)

## Echtzeitanalyse für die Erkennung benutzerdefinierter Entitäten (Konsole)

Sie können die Amazon Comprehend-Konsole verwenden, um Echtzeitanalysen mit einem benutzerdefinierten Modell durchzuführen. Zunächst erstellen Sie einen Endpunkt, um die Echtzeitanalyse auszuführen. Nachdem Sie den Endpunkt erstellt haben, führen Sie die Echtzeitanalyse durch.

Informationen zur Bereitstellung des Endpunktdurchsatzes und zu den damit verbundenen Kosten finden Sie unter [Verwenden von Amazon Comprehend-Endpunkten](#).

#### Themen

- [Erstellen eines Endpunkts für die Erkennung benutzerdefinierter Entitäten](#)
- [Ausführen der Erkennung benutzerdefinierter Entitäten in Echtzeit](#)

## Erstellen eines Endpunkts für die Erkennung benutzerdefinierter Entitäten

So erstellen Sie einen Endpunkt (Konsole)

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon Comprehend-Konsole unter <https://console.aws.amazon.com/comprehend/>
2. Wählen Sie im linken Menü Endpunkte und anschließend die Schaltfläche Endpunkt erstellen aus. Ein Bildschirm Endpunkt erstellen wird geöffnet.
3. Geben Sie dem Endpunkt einen Namen. Der Name muss innerhalb der aktuellen Region und des aktuellen Kontos eindeutig sein.
4. Wählen Sie ein benutzerdefiniertes Modell aus, an das Sie den neuen Endpunkt anfügen möchten. In der Dropdownliste können Sie nach Modellnamen suchen.

### Note

Sie müssen ein Modell erstellen, bevor Sie ihm einen Endpunkt anfügen können. Wenn Sie noch kein Modell haben, finden Sie weitere Informationen unter [Trainieren von benutzerdefinierten Entity Recognizer-Modellen](#).

5. (Optional) Um dem Endpunkt ein Tag hinzuzufügen, geben Sie unter Tags ein Schlüssel-Wert-Paar ein und wählen Sie Tag hinzufügen aus. Um dieses Paar zu entfernen, bevor Sie den Endpunkt erstellen, wählen Sie Tag entfernen aus.
6. Geben Sie die Anzahl der Inferenzeinheiten (IUs) ein, die dem Endpunkt zugewiesen werden sollen. Jede Einheit stellt einen Durchsatz von 100 Zeichen pro Sekunde für bis zu zwei Dokumente pro Sekunde dar. Weitere Informationen zum Endpunktdurchsatz finden Sie unter [Verwenden von Amazon Comprehend-Endpunkten](#).
7. (Optional) Wenn Sie einen neuen Endpunkt erstellen, haben Sie die Möglichkeit, den Bol-Schätzer zu verwenden. Der Schätzer kann Ihnen helfen, die Anzahl der anzufordernden IUs zu bestimmen. Die Anzahl der Inferenzeinheiten hängt vom Durchsatz oder der Anzahl der Zeichen ab, die Sie pro Sekunde analysieren möchten.
8. Überprüfen Sie in der Kaufübersicht Ihre geschätzten stündlichen, täglichen und monatlichen Endpunktkosten.
9. Aktivieren Sie das Kontrollkästchen, wenn Sie verstehen, dass für Ihr Konto ab dem Zeitpunkt, an dem es beginnt, Gebühren für den Endpunkt anfallen, bis Sie ihn löschen.
10. Wählen Sie Endpunkt erstellen aus.

## Ausführen der Erkennung benutzerdefinierter Entitäten in Echtzeit

Nachdem Sie einen Endpunkt für Ihr benutzerdefiniertes Entity Recognizer-Modell erstellt haben, können Sie Echtzeitanalysen durchführen, um Entitäten in einzelnen Dokumenten zu erkennen.

Führen Sie die folgenden Schritte aus, um benutzerdefinierte Entitäten in Ihrem Text mithilfe der Amazon Comprehend-Konsole zu erkennen.

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon Comprehend-Konsole unter <https://console.aws.amazon.com/comprehend/>
2. Wählen Sie im linken Menü Echtzeitanalyse aus.
3. Wählen Sie im Abschnitt Eingabetext für Analysetyp die Option Benutzerdefiniert aus.
4. Wählen Sie für Endpunkt auswählen den Endpunkt aus, der dem Entitätserkennungsmodell zugeordnet ist, das Sie verwenden möchten.
5. Um die Eingabedaten für die Analyse anzugeben, können Sie Text eingeben oder eine Datei hochladen.
  - So geben Sie Text ein:
    - a. Wählen Sie Eingabetext aus.
    - b. Geben Sie den Text ein, den Sie analysieren möchten.
  - So laden Sie eine Datei hoch:
    - a. Wählen Sie Datei hochladen und geben Sie den hochzuladenden Dateinamen ein.
    - b. (Optional) Unter Erweiterte Leseaktionen können Sie die Standardaktionen für die Textextraktion überschreiben. Details hierzu finden Sie unter [Festlegen von Optionen für die Textextraktion](#).
6. Wählen Sie Analysieren aus. Die Konsole zeigt die Ausgabe der Analyse zusammen mit einer Zuverlässigkeitsbewertung an.

## Echtzeitanalyse für die Erkennung benutzerdefinierter Entitäten (API)

Sie können die Amazon Comprehend API verwenden, um Echtzeitanalysen mit einem benutzerdefinierten Modell durchzuführen. Zunächst erstellen Sie einen Endpunkt, um die Echtzeitanalyse auszuführen. Nachdem Sie den Endpunkt erstellt haben, führen Sie die Echtzeitanalyse durch.



Informationen zur Bereitstellung des Endpunktdurchsatzes und zu den damit verbundenen Kosten finden Sie unter [Verwenden von Amazon Comprehend-Endpunkten](#).

## Themen

- [Erstellen eines Endpunkts für die Erkennung benutzerdefinierter Entitäten](#)
- [Ausführen der Erkennung benutzerdefinierter Entitäten in Echtzeit](#)

## Erstellen eines Endpunkts für die Erkennung benutzerdefinierter Entitäten

Informationen zu den Kosten für Endpunkte finden Sie unter [Verwenden von Amazon Comprehend-Endpunkten](#).

### Erstellen eines Endpunkts mit der AWS CLI

Verwenden Sie den `create-endpoint` Befehl AWS CLI, um einen Endpunkt mithilfe der zu erstellen:

```
$ aws comprehend create-endpoint \  
> --desired-inference-units number of inference units \  
> --endpoint-name endpoint name \  
> --model-arn arn:aws:comprehend:region:account-id:model/example \  
> --tags Key=Key,Value=Value
```

Wenn Ihr Befehl erfolgreich ist, antwortet Amazon Comprehend mit dem Endpunkt-ARN:

```
{  
  "EndpointArn": "Arn"  
}
```

Weitere Informationen zu diesem Befehl, seinen Parameterargumenten und seiner Ausgabe finden Sie unter [create-endpoint](#) in der -AWS CLIBefehlsreferenz.

## Ausführen der Erkennung benutzerdefinierter Entitäten in Echtzeit

Nachdem Sie einen Endpunkt für Ihr benutzerdefiniertes Entity Recognizer-Modell erstellt haben, verwenden Sie den Endpunkt, um den [DetectEntities](#) API-Vorgang auszuführen. Sie können Texteingaben mit dem `bytes` Parameter `text` oder bereitstellen. Geben Sie die anderen Eingabetypen mit dem `bytes` Parameter ein.

Für Bilddateien und PDF-Dateien können Sie den `DocumentReaderConfig` Parameter verwenden, um die standardmäßigen Textextraktionsaktionen zu überschreiben. Details hierzu finden Sie unter [Festlegen von Optionen für die Textextraktion](#).

## Erkennen von Entitäten im Text mithilfe der AWS CLI

Um benutzerdefinierte Entitäten im Text zu erkennen, führen Sie den `detect-entities` Befehl mit dem Eingabetext im `text` Parameter aus.

Example : Verwenden der CLI zum Erkennen von Entitäten im Eingabetext

```
$ aws comprehend detect-entities \  
> --endpoint-arn arn \  
> --language-code en \  
> --text "Andy Jassy is the CEO of Amazon."
```

Wenn Ihr Befehl erfolgreich ist, antwortet Amazon Comprehend mit der Analyse. Für jede Entität, die Amazon Comprehend erkennt, gibt es den Entitätstyp, Text, den Standort und den Konfidenzwert an.

## Erkennen von Entitäten in halbstrukturierten Dokumenten mithilfe der AWS CLI

Um benutzerdefinierte Entitäten in PDF-, Wort- oder Bilddateien zu erkennen, führen Sie den `detect-entities` Befehl mit der Eingabedatei im `bytes` Parameter aus.

Example : Verwenden der -CLI zum Erkennen von Entitäten in einer Bilddatei

Dieses Beispiel zeigt, wie die Bilddatei mit der `fileb` Option übergeben wird, um die Bildbytes mit base64 zu codieren. Weitere Informationen finden Sie unter [Binäre große Objekte](#) im AWS Command Line Interface -Benutzerhandbuch.

In diesem Beispiel wird auch eine JSON-Datei mit dem Namen `übergebenconfig.json`, um die Optionen für die Textextraktion festzulegen.

```
$ aws comprehend detect-entities \  
> --endpoint-arn arn \  
> --language-code en \  
> --bytes fileb://image1.jpg \  
> --document-reader-config file://config.json
```

Die Datei config.json enthält den folgenden Inhalt.

```
{
  "DocumentReadMode": "FORCE_DOCUMENT_READ_ACTION",
  "DocumentReadAction": "EXTRACT_DETECT_DOCUMENT_TEXT"
}
```

Weitere Informationen zur Befehlsyntax finden Sie unter [DetectEntities](#) in der Amazon Comprehend API-Referenz.

## Ausgaben für Echtzeitanalysen

### Ausgaben für Texteingaben

Wenn Sie Text mit dem Text Parameter eingeben, besteht die Ausgabe aus einem Array von Entitäten, die die Analyse erkannt hat. Das folgende Beispiel zeigt eine Analyse, die zwei JUDGE-Entitäten erkannt hat.

```
{
  "Entities":
  [
    {
      "BeginOffset": 0,
      "EndOffset": 22,
      "Score": 0.9763959646224976,
      "Text": "John Johnson",
      "Type": "JUDGE"
    },
    {
      "BeginOffset": 11,
      "EndOffset": 15,
      "Score": 0.9615424871444702,
      "Text": "Thomas Kincaid",
      "Type": "JUDGE"
    }
  ]
}
```

## Ausgaben für halbstrukturierte Eingaben

Für ein halbstrukturiertes Eingabedokument oder eine Textdatei kann die Ausgabe die folgenden zusätzlichen Felder enthalten:

- **DocumentMetadata** – Extraktionsinformationen über das Dokument. Die Metadaten enthalten eine Liste von Seiten im Dokument mit der Anzahl der Zeichen, die von jeder Seite extrahiert wurden. Dieses Feld ist in der Antwort vorhanden, wenn die Anforderung den `Byte` Parameter enthält.
- **DocumentType** – Der Dokumenttyp für jede Seite im Eingabedokument. Dieses Feld ist in der Antwort für eine Anforderung vorhanden, die den `Byte` Parameter enthält.
- **Blöcke** – Informationen zu jedem Textblock im Eingabedokument. Blöcke sind verschachtelt. Ein Seitenblock enthält einen Block für jede Textzeile, der einen Block für jedes Wort enthält. Dieses Feld ist in der Antwort für eine Anforderung vorhanden, die den `Byte` Parameter enthält.
- **BlockReferences** – Ein Verweis auf jeden Block für diese Entität. Dieses Feld ist in der Antwort für eine Anforderung vorhanden, die den `Byte` Parameter enthält. Das Feld ist für Textdateien nicht vorhanden.
- **Fehler** – Fehler auf Seitenebene, die das System bei der Verarbeitung des Eingabedokuments erkannt hat. Das Feld ist leer, wenn das System keine Fehler festgestellt hat.

Beschreibungen dieser Ausgabefelder finden Sie unter [DetectEntities](#) in der Amazon Comprehend API-Referenz. Weitere Informationen zu den Layoutelementen finden Sie unter [Amazon Textract-Analyseobjekte](#) im Amazon Textract-Entwicklerhandbuch.

Das folgende Beispiel zeigt die Ausgabe für ein einseitiges gescanntes PDF-Eingabedokument.

```
{
  "Entities": [{
    "Score": 0.9984670877456665,
    "Type": "DATE-TIME",
    "Text": "September 4,",
    "BlockReferences": [{
      "BlockId": "42dcaae-c484-4b5d-9e3f-ae0be928b3e1",
      "BeginOffset": 0,
      "EndOffset": 12,
      "ChildBlocks": [{
        "ChildBlockId": "6e9cbb43-f8be-4da0-9a4b-ff9a6c350a14",
        "BeginOffset": 0,
        "EndOffset": 9
      }],
    }],
  }],
}
```

```
        {
            "ChildBlockId": "599e0d53-ae9f-491b-a762-459b22c79ff5",
            "BeginOffset": 0,
            "EndOffset": 2
        },
        {
            "ChildBlockId": "599e0d53-ae9f-491b-a762-459b22c79ff5",
            "BeginOffset": 0,
            "EndOffset": 2
        }
    ]
}]]
}],
"DocumentMetadata": {
    "Pages": 1,
    "ExtractedCharacters": [{
        "Page": 1,
        "Count": 609
    }]
},
"DocumentType": [{
    "Page": 1,
    "Type": "SCANNED_PDF"
}],
"Blocks": [{
    "Id": "ee82edf3-28de-4d63-8883-40e2e4938ccb",
    "BlockType": "LINE",
    "Text": "Your Band",
    "Page": 1,
    "Geometry": {
        "BoundingBox": {
            "Height": 0.024125460535287857,
            "Left": 0.11745482683181763,
            "Top": 0.06821706146001816,
            "Width": 0.12074867635965347
        },
        "Polygon": [{
            "X": 0.11745482683181763,
            "Y": 0.06821706146001816
        },
        {
            "X": 0.2382034957408905,
            "Y": 0.06821706146001816
        }
    ]
},
```

```

    {
      "X": 0.2382034957408905,
      "Y": 0.09234252572059631
    },
    {
      "X": 0.11745482683181763,
      "Y": 0.09234252572059631
    }
  ]
},
"Relationships": [{
  "Ids": [
    "b105c561-c8d9-485a-a728-7a5b1a308935",
    "60ecb119-3173-4de2-8c5d-de182a5f86a5"
  ],
  "Type": "CHILD"
}]
}]
}

```

Das folgende Beispiel zeigt die Ausgabe für die Analyse eines nativen PDF-Dokuments.

Example Beispielausgabe einer benutzerdefinierten Entitätserkennungsanalyse eines PDF-Dokuments

```

{
  "Blocks":
  [
    {
      "BlockType": "LINE",
      "Geometry":
      {
        "BoundingBox":
        {
          "Height": 0.012575757575757575,
          "Left": 0.0,
          "Top": 0.0015063131313131314,
          "Width": 0.02262091503267974
        },
        "Polygon":
        [
          {
            "X": 0.0,

```

```
        "Y": 0.0015063131313131314
      },
      {
        "X": 0.02262091503267974,
        "Y": 0.0015063131313131314
      },
      {
        "X": 0.02262091503267974,
        "Y": 0.014082070707070706
      },
      {
        "X": 0.0,
        "Y": 0.014082070707070706
      }
    ]
  },
  "Id": "4330efed-6334-4fc4-ba48-e050afa95c8d",
  "Page": 1,
  "Relationships":
  [
    {
      "ids":
      [
        "f343ce48-583d-4abe-b84b-a232e266450f"
      ],
      "type": "CHILD"
    }
  ],
  "Text": "S-3"
},
{
  "BlockType": "WORD",
  "Geometry":
  {
    "BoundingBox":
    {
      "Height": 0.012575757575757575,
      "Left": 0.0,
      "Top": 0.0015063131313131314,
      "Width": 0.02262091503267974
    },
    "Polygon":
    [
      {
```

```
        "X": 0.0,
        "Y": 0.0015063131313131314
    },
    {
        "X": 0.02262091503267974,
        "Y": 0.0015063131313131314
    },
    {
        "X": 0.02262091503267974,
        "Y": 0.014082070707070706
    },
    {
        "X": 0.0,
        "Y": 0.014082070707070706
    }
]
},
"Id": "f343ce48-583d-4abe-b84b-a232e266450f",
"Page": 1,
"Relationships":
[],
"Text": "S-3"
}
],
"DocumentMetadata":
{
    "PageNumber": 1,
    "Pages": 1
},
"DocumentType": "NativePDF",
"Entities":
[
    {
        "BlockReferences":
        [
            {
                "BeginOffset": 25,
                "BlockId": "4330efed-6334-4fc4-ba48-e050afa95c8d",
                "ChildBlocks":
                [
                    {
                        "BeginOffset": 1,
                        "ChildBlockId": "cbba5534-ac69-4bc4-beef-306c659f70a6",
                        "EndOffset": 6
                    }
                ]
            }
        ]
    }
]
```



```
        }
      ],
      "EndOffset": 30
    }
  ],
  "Score": 0.9998825926329088,
  "Text": "0.001",
  "Type": "OFFERING_PRICE"
},
{
  "BlockReferences":
  [
    {
      "BeginOffset": 41,
      "BlockId": "f343ce48-583d-4abe-b84b-a232e266450f",
      "ChildBlocks":
      [
        {
          "BeginOffset": 0,
          "ChildBlockId": "292a2e26-21f0-401b-a2bf-03aa4c47f787",
          "EndOffset": 9
        }
      ],
      "EndOffset": 50
    }
  ],
  "Score": 0.9809727537330395,
  "Text": "6,097,560",
  "Type": "OFFERED_SHARES"
}
],
"File": "example.pdf",
"Version": "2021-04-30"
}
```

## Ausführen von Analyseaufträgen zur Erkennung benutzerdefinierter Entitäten

Sie können einen asynchronen Analyseauftrag ausführen, um benutzerdefinierte Entitäten in einem Satz von einem oder mehreren Dokumenten zu erkennen.

### Bevor Sie beginnen

Sie benötigen ein benutzerdefiniertes Entitätserkennungsmodell (auch bekannt als Recognizer), bevor Sie benutzerdefinierte Entitäten erkennen können. Weitere Informationen zu diesen Modellen finden Sie unter [the section called “Trainingserkennungsmodelle”](#).

Ein Recognizer, der mit Klartext-Anmerkungen trainiert wurde, unterstützt nur die Entitätserkennung für Klartextdokumente. Ein Recognizer, der mit Anmerkungen zu PDF-Dokumenten trainiert wurde, unterstützt die Entitätserkennung für Klartextdokumente, Bilder, PDF-Dateien und Word-Dokumente. Für andere Dateien als Textdateien führt Amazon Comprehend eine Textextraktion durch, bevor die Analyse ausgeführt wird. Informationen zu den Eingabedateien finden Sie unter [Eingaben für asynchrone benutzerdefinierte Analysen](#).

Wenn Sie planen, Bilddateien oder gescannte PDF-Dokumente zu analysieren, muss Ihre IAM-Richtlinie Berechtigungen zur Verwendung von zwei Amazon Textract-API-Methoden (DetectDocumentText und AnalyzeDocument) erteilen. Amazon Comprehend ruft diese Methoden während der Textextraktion auf. Eine Beispielrichtlinie finden Sie unter [Erforderliche Berechtigungen zum Ausführen von Dokumentanalyseaktionen](#).

Um einen asynchronen Analyseauftrag auszuführen, führen Sie die folgenden allgemeinen Schritte aus:

1. Speichern Sie die Dokumente in einem Amazon S3-Bucket.
2. Verwenden Sie die API oder Konsole, um den Analyseauftrag zu starten.
3. Überwachen Sie den Fortschritt des Analyseauftrags.
4. Nachdem der Auftrag abgeschlossen ist, rufen Sie die Ergebnisse der Analyse aus dem S3-Bucket ab, den Sie beim Starten des Auftrags angegeben haben.

### Themen

- [Starten eines benutzerdefinierten Entitätserkennungsauftrags \(Konsole\)](#)
- [Starten eines benutzerdefinierten Entitätserkennungsauftrags \(API\)](#)
- [Ausgaben für asynchrone Analyseaufträge](#)

## Starten eines benutzerdefinierten Entitätserkennungsauftrags (Konsole)

Sie können die Konsole verwenden, um einen asynchronen Analyseauftrag für die Erkennung benutzerdefinierter Entitäten zu starten und zu überwachen.

So starten Sie einen asynchronen Analyseauftrag


1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon Comprehend-Konsole unter <https://console.aws.amazon.com/comprehend/>
2. Wählen Sie im linken Menü Analyseaufträge und dann Auftrag erstellen aus.
3. Geben Sie dem Klassifizierungsauftrag einen Namen. Der Name muss in Ihrem Konto und in der aktuellen Region eindeutig sein.
4. Wählen Sie unter Analysetyp die Option Benutzerdefinierte Entitätserkennung aus.
5. Wählen Sie unter Recognizer model die zu verwendende benutzerdefinierte Entitätserkennung aus.
6. Wählen Sie unter Version die zu verwendende Erkennungsversion aus.
7. (Optional) Wenn Sie die Daten verschlüsseln möchten, die Amazon Comprehend bei der Verarbeitung Ihres Auftrags verwendet, wählen Sie Auftragsverschlüsselung aus. Wählen Sie dann aus, ob Sie einen KMS-Schlüssel verwenden möchten, der dem aktuellen Konto zugeordnet ist, oder einen Schlüssel aus einem anderen Konto.
  - Wenn Sie einen Schlüssel verwenden, der dem aktuellen Konto zugeordnet ist, wählen Sie die Schlüssel-ID für die KMS-Schlüssel-ID aus.
  - Wenn Sie einen Schlüssel verwenden, der einem anderen Konto zugeordnet ist, geben Sie den ARN für die Schlüssel-ID unter KMS-Schlüssel-ARN ein.

### Note

Weitere Informationen zum Erstellen und Verwenden von KMS-Schlüsseln und der zugehörigen Verschlüsselung finden Sie unter [Key Management Service \(KMS\)](#).

8. Geben Sie unter Eingabedaten den Speicherort des Amazon S3-Buckets ein, der Ihre Eingabedokumente enthält, oder navigieren Sie zu diesem, indem Sie S3 durchsuchen auswählen. Dieser Bucket muss sich in derselben Region befinden wie die API, die Sie aufrufen. Die IAM-Rolle, die Sie für Zugriffsberechtigungen für den Analyseauftrag verwenden, muss über Leseberechtigungen für den S3-Bucket verfügen.

9. (Optional) Für Eingabeformat können Sie das Format der Eingabedokumente auswählen. Das Format kann ein Dokument pro Datei oder ein Dokument pro Zeile in einer einzigen Datei sein. Ein Dokument pro Zeile gilt nur für Textdokumente.
10. (Optional) Für den Dokumentlesemodus können Sie die standardmäßigen Textextraktionsaktionen überschreiben. Weitere Informationen finden Sie unter [Festlegen von Optionen für die Textextraktion](#).
11. Geben Sie unter Ausgabedaten den Speicherort des Amazon S3-Buckets ein, in den Amazon Comprehend die Ausgabedaten des Auftrags schreiben soll, oder navigieren Sie zu diesem, indem Sie S3 durchsuchen auswählen. Dieser Bucket muss sich in derselben Region befinden wie die API, die Sie aufrufen. Die IAM-Rolle, die Sie für Zugriffsberechtigungen für den Klassifizierungsauftrag verwenden, muss über Schreibberechtigungen für den S3-Bucket verfügen.
12. (Optional) Wenn Sie das Ausgabeergebnis Ihres Auftrags verschlüsseln möchten, wählen Sie Verschlüsselung aus. Wählen Sie dann aus, ob Sie einen KMS-Schlüssel verwenden möchten, der dem aktuellen Konto zugeordnet ist, oder einen Schlüssel aus einem anderen Konto.
  - Wenn Sie einen Schlüssel verwenden, der dem aktuellen Konto zugeordnet ist, wählen Sie den Schlüsselalias oder die ID für die KMS-Schlüssel-ID aus.
  - Wenn Sie einen Schlüssel verwenden, der einem anderen Konto zugeordnet ist, geben Sie den ARN für den Schlüsselalias oder die ID unter KMS-Schlüssel-ID ein.
13. (Optional) Um Ihre Ressourcen in Amazon Comprehend von einer VPC aus zu starten, geben Sie die VPC-ID unter VPC ein oder wählen Sie die ID aus der Dropdown-Liste aus.
  1. Wählen Sie das Subnetz unter Subnetz(e) aus. Nachdem Sie das erste Subnetz ausgewählt haben, können Sie weitere auswählen.
  2. Wählen Sie unter Sicherheitsgruppe(n) die zu verwendende Sicherheitsgruppe aus, falls Sie eine angegeben haben. Nachdem Sie die erste Sicherheitsgruppe ausgewählt haben, können Sie zusätzliche auswählen.

 Note

Wenn Sie eine VPC mit Ihrem Analyseauftrag verwenden, muss die für die Erstellungs- und Startvorgänge `DataAccessRole` verwendete über Berechtigungen für die VPC verfügen, die auf den Ausgabe-Bucket zugreift.

14. Wählen Sie Auftrag erstellen, um den Entitätserkennungsauftrag zu erstellen.

## Starten eines benutzerdefinierten Entitätserkennungsauftrags (API)

Sie können die `-API` verwenden, um einen asynchronen Analyseauftrag für die Erkennung benutzerdefinierter Entitäten zu starten und zu überwachen.

Um einen benutzerdefinierten Entitätserkennungsauftrag mit der `-StartEntitiesDetectionJob` Operation zu starten, geben Sie den `EntityRecognizerArn` Amazon-Ressourcennamen (ARN) des trainierten Modells an. Sie finden diesen ARN in der Antwort auf die `-CreateEntityRecognizer` Operation.

### Themen

- [Erkennen von benutzerdefinierten Entitäten mithilfe der AWS Command Line Interface](#)
- [Erkennen von benutzerdefinierten Entitäten mithilfe der AWS SDK for Java](#)
- [Erkennen von benutzerdefinierten Entitäten mithilfe der AWS SDK for Python \(Boto3\)](#)
- [Überschreiben von API-Aktionen für PDF-Dateien](#)

## Erkennen von benutzerdefinierten Entitäten mithilfe der AWS Command Line Interface

Verwenden Sie das folgende Beispiel für Unix-, Linux- und macOS-Umgebungen. Ersetzen Sie unter Windows den umgekehrten Schrägstrich (`\`), das Unix-Fortsetzungszeichen, am Ende jeder Zeile durch ein Caret-Zeichen oder Zirkumflex (`^`). Verwenden Sie die folgende Anforderungssyntax, um benutzerdefinierte Entitäten in einem Dokumentsatz zu erkennen:

```
aws comprehend start-entities-detection-job \  
  --entity-recognizer-arn "arn:aws:comprehend:region:account number:entity-  
recognizer/test-6" \  
  --job-name infer-1 \  
  --data-access-role-arn "arn:aws:iam::account number:role/service-role/  
AmazonComprehendServiceRole-role" \  
  --language-code en \  
  --input-data-config "S3Uri=s3://Bucket Name/Bucket Path" \  
  --output-data-config "S3Uri=s3://Bucket Name/Bucket Path/" \  
  --region region
```

Amazon Comprehend antwortet mit `JobID` `JobStatus` und gibt die Ausgabe des Auftrags in dem S3-Bucket zurück, den Sie in der Anforderung angegeben haben.

## Erkennen von benutzerdefinierten Entitäten mithilfe der AWS SDK for Java

Beispiele für Amazon Comprehend, die Java verwenden, finden Sie unter [Amazon Comprehend Java-Beispiele](#).

## Erkennen von benutzerdefinierten Entitäten mithilfe der AWS SDK for Python (Boto3)

In diesem Beispiel wird ein benutzerdefinierter Entity Recognizer erstellt, das Modell trainiert und dann in einem Entity Recognizer-Auftrag mit der ausgeführt AWS SDK for Python (Boto3).

Instanzieren Sie das SDK für Python.

```
import boto3
import uuid
comprehend = boto3.client("comprehend", region_name="region")
```

Erstellen Sie einen Entity Recognizer:

```
response = comprehend.create_entity_recognizer(
    RecognizerName="Recognizer-Name-Goes-Here-{}".format(str(uuid.uuid4())),
    LanguageCode="en",
    DataAccessRoleArn="Role ARN",
    InputDataConfig={
        "EntityTypes": [
            {
                "Type": "ENTITY_TYPE"
            }
        ],
        "Documents": {
            "S3Uri": "s3://Bucket Name/Bucket Path/documents"
        },
        "Annotations": {
            "S3Uri": "s3://Bucket Name/Bucket Path/annotations"
        }
    }
)
recognizer_arn = response["EntityRecognizerArn"]
```

Listet alle Erkennungen auf:

```
response = comprehend.list_entity_recognizers()
```

Warten Sie, bis der Entity Recognizer den Status TRAINED erreicht:

```
while True:
    response = comprehend.describe_entity_recognizer(
        EntityRecognizerArn=recognizer_arn
    )

    status = response["EntityRecognizerProperties"]["Status"]
    if "IN_ERROR" == status:
        sys.exit(1)
    if "TRAINED" == status:
        break

    time.sleep(10)
```

Starten Sie einen Auftrag zur Erkennung benutzerdefinierter Entitäten:

```
response = comprehend.start_entities_detection_job(
    EntityRecognizerArn=recognizer_arn,
    JobName="Detection-Job-Name-{}".format(str(uuid.uuid4())),
    LanguageCode="en",
    DataAccessRoleArn="Role ARN",
    InputDataConfig={
        "InputFormat": "ONE_DOC_PER_LINE",
        "S3Uri": "s3://Bucket Name/Bucket Path/documents"
    },
    OutputDataConfig={
        "S3Uri": "s3://Bucket Name/Bucket Path/output"
    }
)
```

## Überschreiben von API-Aktionen für PDF-Dateien

Für Bilddateien und PDF-Dateien können Sie die Standard-Extraktionsaktionen mit dem `DocumentReaderConfig` Parameter in `überschreibenInputDataConfig`.

Im folgenden Beispiel wird eine JSON-Datei mit dem Namen `myInputDataConfig.json` definiert, um die `InputDataConfig` Werte festzulegen. Sie legt fest `DocumentReadConfig`, dass die Amazon Textract `DetectDocumentText` API für alle PDF-Dateien verwendet wird.

## Example

```
"InputDataConfig": {
  "S3Uri": s3://Bucket Name/Bucket Path",
  "InputFormat": "ONE_DOC_PER_FILE",
  "DocumentReaderConfig": {
    "DocumentReadAction": "EXTRACT_DETECT_DOCUMENT_TEXT",
    "DocumentReadMode": "FORCE_DOCUMENT_READ_ACTION"
  }
}
```

Geben Sie in der `-StartEntitiesDetectionJobOperation` die Datei `myInputDataConfig.json` als `InputDataConfig` Parameter an:

```
--input-data-config file://myInputDataConfig.json
```

Weitere Informationen zu den `DocumentReaderConfig` Parametern finden Sie unter [Festlegen von Optionen für die Textextraktion](#).

## Ausgaben für asynchrone Analyseaufträge

Nachdem ein Analyseauftrag abgeschlossen ist, speichert er die Ergebnisse in dem S3-Bucket, den Sie in der Anforderung angegeben haben.

### Ausgaben für Texteingaben

Bei Texteingabedateien besteht die Ausgabe aus einer Liste von Entitäten für jedes Eingabedokument.

Das folgende Beispiel zeigt die Ausgabe für zwei Dokumente aus einer Eingabedatei mit dem Namen `50_docs`, wobei ein Dokument pro Zeilenformat verwendet wird.

```
{
  "File": "50_docs",
  "Line": 0,
  "Entities":
  [
    {
      "BeginOffset": 0,
      "EndOffset": 22,
```



```
        "Score": 0.9763959646224976,  
        "Text": "John Johnson",  
        "Type": "JUDGE"  
    }  
]  
}  
{  
  "File": "50_docs",  
  "Line": 1,  
  "Entities":  
  [  
    {  
      "BeginOffset": 11,  
      "EndOffset": 15,  
      "Score": 0.9615424871444702,  
      "Text": "Thomas Kincaid",  
      "Type": "JUDGE"  
    }  
  ]  
}
```

## Ausgaben für halbstrukturierte Eingaben

Bei halbstrukturierten Eingabedokumenten kann die Ausgabe die folgenden zusätzlichen Felder enthalten:

- **DocumentMetadata** – Extraktionsinformationen über das Dokument. Die Metadaten enthalten eine Liste von Seiten im Dokument mit der Anzahl der von jeder Seite extrahierten Zeichen. Dieses Feld ist in der Antwort vorhanden, wenn die Anforderung den `Byte` Parameter enthielt.
- **DocumentType** – Der Dokumenttyp für jede Seite im Eingabedokument. Dieses Feld ist in der Antwort für eine Anforderung vorhanden, die den `Byte` Parameter enthält.
- **Blöcke** – Informationen zu jedem Textblock im Eingabedokument. Blöcke können innerhalb eines Blocks verschachtelt werden. Ein Seitenblock enthält einen Block für jede Textzeile, der einen Block für jedes Wort enthält. Dieses Feld ist in der Antwort für eine Anforderung vorhanden, die den `Byte` Parameter enthält.
- **BlockReferences** – Ein Verweis auf jeden Block für diese Entität. Dieses Feld ist in der Antwort für eine Anforderung vorhanden, die den `Byte` Parameter enthält. Das Feld ist für Textdateien nicht vorhanden.
- **Fehler** – Fehler auf Seitenebene, die das System bei der Verarbeitung des Eingabedokuments erkannt hat. Das Feld ist leer, wenn das System keine Fehler festgestellt hat.

Weitere Informationen zu diesen Ausgabefeldern finden Sie unter [DetectEntities](#) in der Amazon Comprehend API-Referenz.

Das folgende Beispiel zeigt die Ausgabe für ein einseitiges natives PDF-Eingabedokument.

Example Beispielausgabe einer benutzerdefinierten Entitätserkennungsanalyse eines PDF-Dokuments

```
{
  "Blocks":
  [
    {
      "BlockType": "LINE",
      "Geometry":
      {
        "BoundingBox":
        {
          "Height": 0.012575757575757575,
          "Left": 0.0,
          "Top": 0.0015063131313131314,
          "Width": 0.02262091503267974
        },
        "Polygon":
        [
          {
            "X": 0.0,
            "Y": 0.0015063131313131314
          },
          {
            "X": 0.02262091503267974,
            "Y": 0.0015063131313131314
          },
          {
            "X": 0.02262091503267974,
            "Y": 0.014082070707070706
          },
          {
            "X": 0.0,
            "Y": 0.014082070707070706
          }
        ]
      },
      "Id": "4330efed-6334-4fc4-ba48-e050afa95c8d",
    }
  ]
}
```

```
"Page": 1,
"Relationships":
[
  {
    "ids":
    [
      "f343ce48-583d-4abe-b84b-a232e266450f"
    ],
    "type": "CHILD"
  }
],
"Text": "S-3"
},
{
  "BlockType": "WORD",
  "Geometry":
  {
    "BoundingBox":
    {
      "Height": 0.012575757575757575,
      "Left": 0.0,
      "Top": 0.0015063131313131314,
      "Width": 0.02262091503267974
    },
    "Polygon":
    [
      {
        "X": 0.0,
        "Y": 0.0015063131313131314
      },
      {
        "X": 0.02262091503267974,
        "Y": 0.0015063131313131314
      },
      {
        "X": 0.02262091503267974,
        "Y": 0.014082070707070706
      },
      {
        "X": 0.0,
        "Y": 0.014082070707070706
      }
    ]
  }
},
```

```
        "Id": "f343ce48-583d-4abe-b84b-a232e266450f",
        "Page": 1,
        "Relationships":
        [],
        "Text": "S-3"
    }
],
"DocumentMetadata":
{
    "PageNumber": 1,
    "Pages": 1
},
"DocumentType": "NativePDF",
"Entities":
[
    {
        "BlockReferences":
        [
            {
                "BeginOffset": 25,
                "BlockId": "4330efed-6334-4fc4-ba48-e050afa95c8d",
                "ChildBlocks":
                [
                    {
                        "BeginOffset": 1,
                        "ChildBlockId": "cbba5534-ac69-4bc4-beef-306c659f70a6",
                        "EndOffset": 6
                    }
                ],
                "EndOffset": 30
            }
        ],
        "Score": 0.9998825926329088,
        "Text": "0.001",
        "Type": "OFFERING_PRICE"
    },
    {
        "BlockReferences":
        [
            {
                "BeginOffset": 41,
                "BlockId": "f343ce48-583d-4abe-b84b-a232e266450f",
                "ChildBlocks":
                [
```

```
        {
            "BeginOffset": 0,
            "ChildBlockId": "292a2e26-21f0-401b-a2bf-03aa4c47f787",
            "EndOffset": 9
        }
    ],
    "EndOffset": 50
}
],
"Score": 0.9809727537330395,
"Text": "6,097,560",
"Type": "OFFERED_SHARES"
}
],
"File": "example.pdf",
"Version": "2021-04-30"
}
```

# Erstellen und Verwalten von benutzerdefinierten Modellen

Amazon Comprehend enthält integrierte NLP-Modelle (natürliche Sprachverarbeitung), mit denen Sie Erkenntnisse oder Themenmodellierung analysieren können. Sie können Amazon Comprehend auch verwenden, um benutzerdefinierte Modelle für die Entitätserkennung und Dokumentenklassifizierung zu erstellen.

Sie können die Modellversionsverwaltung verwenden, um den Verlauf Ihres Modells zu verfolgen. Wenn Sie eine neue Modellversion erstellen und trainieren, können Sie Änderungen am Trainingsdatensatz vornehmen. Amazon Comprehend zeigt Details (einschließlich Modellleistung) für jede Modellversion auf der Seite mit den Modelldetails an. Im Laufe der Zeit können Sie sehen, wie sich die Modellleistung ändert, wenn Sie Änderungen an Ihrem Trainingsdatensatz vornehmen.

Sie können Modellversionen mithilfe der Amazon Comprehend-Konsole oder API erstellen. Als Alternative bietet Amazon Comprehend , [Flywheels](#) um die Aufgaben im Zusammenhang mit dem Training und der Bewertung neuer benutzerdefinierter Modellversionen zu vereinfachen.

Nachdem Sie ein benutzerdefiniertes Modell erstellt haben, können Sie das Modell mit anderen Benutzern teilen, indem Sie anderen erlauben AWS-Konten, eine Kopie Ihres Modells zu importieren.

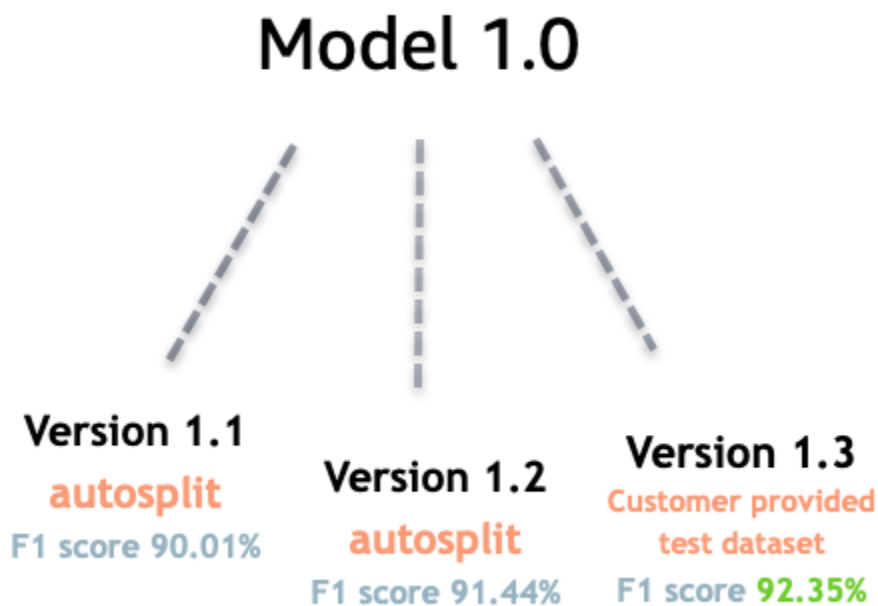
## Themen

- [Modell-Versioning mit Amazon Comprehend](#)
- [Kopieren von benutzerdefinierten Modellen zwischen AWS-Konten](#)

## Modell-Versioning mit Amazon Comprehend

Beim Thema künstliche Intelligenz und Machine Learning (KI/ML) geht es um schnelle Experimente. Mit Amazon Comprehend schulen und erstellen Sie Modelle, mit denen Sie Einblicke in Ihre Daten erhalten. Mit der Modell-Versionsverwaltung können Sie Ihren Modellierungsverlauf und Ihre Ergebnisse im Zusammenhang mit den laufenden Ergebnissen Ihrer Modelle verfolgen, während Sie mehr oder verschiedene Datensätze bereitstellen. Sie können Versioning mit Ihren benutzerdefinierten Klassifizierungsmodellen oder Ihren benutzerdefinierten Entitätserkennungsmodellen verwenden. Wenn Sie sich Ihre verschiedenen Versionen im Laufe der Zeit ansehen, können Sie sich einen Überblick darüber verschaffen, wie erfolgreich sie ausgeführt wurden, und einen Überblick darüber erhalten, welche Parameter Sie verwendet haben, um Ihren Erfolg zu erreichen.

Wenn Sie eine neue Version eines vorhandenen benutzerdefinierten Classifier-Modells oder Entitätserkennungsmodells trainieren, müssen Sie lediglich eine neue Version auf der Modelldetailseite erstellen und alle Details werden für Sie ausgefüllt. Die neue Version hat denselben Namen wie Ihr früheres Modell – was wir `versionID` nennen – obwohl Sie ihr bei der Erstellung einen eindeutigen Versionsnamen geben. Wenn Sie einem Modell neue Versionen hinzufügen, können Sie alle vorherigen Versionen und ihre Details in einer Ansicht auf der Seite mit den Modelldetails sehen. Mit Versioning können Sie sehen, wie sich die Modellleistung ändert, wenn Sie Änderungen an Ihrem Trainingsdatensatz vornehmen.



Erstellen einer neuen benutzerdefinierten Classifier-Version (Konsole)

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon Comprehend-Konsole unter <https://console.aws.amazon.com/comprehend/>
2. Wählen Sie im linken Menü Anpassen und dann Benutzerdefinierte Klassifizierung aus.
3. Wählen Sie in der Liste Classifiers den Namen des benutzerdefinierten Modells aus, aus dem Sie eine neue Version erstellen möchten. Die Seite mit den benutzerdefinierten Modelldetails wird angezeigt.

4. Wählen Sie oben rechts Neues Modell erstellen aus. Es wird ein Bildschirm mit vorausgefüllten Details aus dem übergeordneten benutzerdefinierten Klassifizierungsmodell geöffnet.
5. Fügen Sie unter Versionsname der neuen Version einen eindeutigen Namen hinzu.
6. Unter Versionsdetails können Sie die Sprache und Anzahl der Labels ändern, die Ihrem neuen Modell zugeordnet sind.
7. Konfigurieren Sie im Abschnitt Datenspezifikationen, wie Sie die Daten für Ihre neue Version bereitstellen möchten. Stellen Sie sicher, dass Sie vollständige Daten bereitstellen, die Dokumente aus Ihrem vorherigen Modell und Ihre neuen Dokumente enthalten. Sie können den Classifier-Modus (Single Label oder Multi Label), das Datenformat (CSV-Datei, Augmented Manifest), Ihren Trainingsdatensatz und Ihren Testdatensatz (Autosplit oder Ihre benutzerdefinierte Testdatenkonfiguration) ändern.
8. (Optional) Aktualisieren des S3-Speicherorts für Ihre Ausgabedaten
9. Erstellen oder verwenden Sie unter Zugriffsberechtigungen eine vorhandene IAM-Rolle.
10. (Optional) Aktualisieren Ihrer VPC-Einstellungen
11. (Optional) Fügen Sie Ihrer neuen Version Tags hinzu, um die Details zu verfolgen.

Weitere Informationen zum Erstellen von benutzerdefinierten Classifiern finden Sie unter [Erstellen eines benutzerdefinierten Classifiers](#).

#### Erstellen einer neuen Custom Entity Recognizer-Version (Konsole)

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon Comprehend-Konsole unter <https://console.aws.amazon.com/comprehend/>
2. Wählen Sie im linken Menü Anpassen und dann Benutzerdefinierte Entitätserkennung aus.
3. Wählen Sie in der Liste Recognizer model den Namen des Recognizers aus, aus dem Sie eine neue Version erstellen möchten. Die Detailseite wird angezeigt.
4. Wählen Sie oben rechts Neue Version trainieren aus. Ein Bildschirm mit vorausgefüllten Details von der übergeordneten Entitätserkennung wird geöffnet.
5. Fügen Sie unter Versionsname der neuen Version einen eindeutigen Namen hinzu.
6. Fügen Sie unter Benutzerdefinierter Entitätstyp die benutzerdefinierten Labels oder Labels hinzu, die der Erkenner in Ihrem Datensatz identifizieren soll, und wählen Sie Typ hinzufügen aus. Wählen Sie einen benutzerdefinierten Entitätstyp aus den Anmerkungen oder der Entitätsliste aus, die Sie angegeben haben. Der Recognizer verwendet dann alle enthaltenen Entitätstypen, um Entitäten im Datensatz zu identifizieren, wenn er Ihren Auftrag ausführt. Jeder Entitätstyp



muss in Großbuchstaben geschrieben und durch Unterstrich getrennt werden, wenn er mehrere Wörter verwendet. Es sind maximal 25 Typen zulässig.

7. (Optional) Wählen Sie Recognizer encryption aus, um die Daten im Speichervolume zu verschlüsseln, während Ihr Auftrag verarbeitet wird.
8. Geben Sie im Abschnitt Trainingsdaten die Details zu Annotation und Datenformat (CSV-Datei, erweitertes Manifest) Single-Label oder Multi-Label), das Datenformat (CSV, erweitertes Manifest), Ihren Trainingsdatensatz und Ihren Testdatensatz (autosplit oder Ihre benutzerdefinierte Testdatenkonfiguration) an.
9. (Optional) Aktualisieren des S3-Speicherorts für Ihre Ausgabedaten
10. Erstellen oder verwenden Sie unter Zugriffsberechtigungen eine vorhandene IAM-Rolle.
11. (Optional) Aktualisieren Ihrer VPC-Einstellungen
12. (Optional) Fügen Sie Ihrer neuen Version Tags hinzu, um die Details zu verfolgen.

Weitere Informationen zu benutzerdefinierten Entitätserkennungen finden Sie unter [Benutzerdefinierte Entitätserkennung](#) und [Erstellen eines benutzerdefinierten Entitätserkennungsdienstes mithilfe der Konsole](#).

## Kopieren von benutzerdefinierten Modellen zwischen AWS-Konten

Amazon Comprehend-Benutzer können trainierte benutzerdefinierte Modelle AWS-Konten in einem zweistufigen Prozess zwischen kopieren. Zunächst gibt ein Benutzer in einem AWS-Konto (Konto A) ein benutzerdefiniertes Modell frei, das sich in seinem Konto befindet. Anschließend importiert ein Benutzer in einem anderen AWS-Konto (Konto B) das Modell in sein Konto. Der Benutzer von Konto B muss das Modell nicht trainieren und muss weder die ursprünglichen Trainingsdaten noch die Testdaten kopieren (oder darauf zugreifen).

Um ein benutzerdefiniertes Modell in Konto A freizugeben, fügt der Benutzer einer Modellversion eine AWS Identity and Access Management (IAM)-Richtlinie an. Diese Richtlinie autorisiert eine Entität in Konto B, z. B. einen Benutzer oder eine Rolle, die Modellversion in Amazon Comprehend in ihrem zu importieren AWS-Konto. Der Benutzer von Konto B muss das Modell in dasselbe importieren AWS-Region wie das ursprüngliche Modell.

Um das Modell in Konto B zu importieren, stellt der Benutzer dieses Kontos Amazon Comprehend die erforderlichen Details bereit, z. B. den Amazon-Ressourcennamen (ARN) des Modells. Durch den Import des Modells erstellt dieser Benutzer ein neues benutzerdefiniertes Modell in seinem

AWS-Konto, das das importierte Modell repliziert. Dieses Modell ist vollständig trainiert und für Inferenzaufträge bereit, z. B. zur Dokumentenklassifizierung oder zur Erkennung benannter Entitäten.

Das Kopieren eines benutzerdefinierten Modells ist nützlich, wenn:

- Sie gehören zu einer Organisation, die mehrere verwendet AWS-Konten. Beispielsweise kann Ihre Organisation AWS-Konto für jede Entwicklungsphase, z. B. Aufbau, Phase, Test und Bereitstellung, über einen verfügen. Oder sie kann AWS-Konten sich für geschäftliche Funktionen wie Datenwissenschaft und Technik unterscheiden.
- Ihre Organisation arbeitet mit einem anderen, z. B. einem -AWS-Partner, der benutzerdefinierte Modelle in Amazon Comprehend trainiert und Ihnen als Client zur Verfügung stellt.

In solchen Szenarien können Sie schnell einen trainierten benutzerdefinierten Entity Recognizer oder Dokumentklassifizierer von einer AWS-Konto in eine andere kopieren. Das Kopieren eines Modells auf diese Weise ist einfacher als die Alternative, bei der Sie Trainingsdaten zwischen kopieren, AWS-Konten um doppelte Modelle zu trainieren.

## Themen

- [Freigeben eines benutzerdefinierten Modells für ein anderes AWS-Konto](#)
- [Importieren eines benutzerdefinierten Modells aus einem anderen AWS-Konto](#)

## Freigeben eines benutzerdefinierten Modells für ein anderes AWS-Konto

Mit Amazon Comprehend können Sie Ihre benutzerdefinierten Modelle mit anderen teilen, sodass diese Ihre Modelle in ihre AWS Konten importieren können. Wenn ein Benutzer eines Ihrer benutzerdefinierten Modelle importiert, erstellt er ein neues benutzerdefiniertes Modell in seinem Konto. Ihr neues Modell dupliziert das Modell, das Sie geteilt haben.

Um ein benutzerdefiniertes Modell freizugeben, fügen Sie ihm eine Richtlinie an, die andere zum Importieren autorisiert. Anschließend stellen Sie diesen Benutzern die Details bereit, die sie benötigen.

### Note

Wenn andere Benutzer ein benutzerdefiniertes Modell importieren, das Sie freigegeben haben, müssen sie dasselbe verwenden, z. AWS-Region B. USA Ost (Nord-Virginia), das Ihr Modell enthält.

## Themen

- [Bevor Sie beginnen](#)
- [Ressourcenbasierte Richtlinien für benutzerdefinierte Modelle](#)
- [Schritt 1: Hinzufügen einer ressourcenbasierten Richtlinie zu einem benutzerdefinierten Modell](#)
- [Schritt 2: Geben Sie die Details an, die andere importieren müssen](#)

## Bevor Sie beginnen

Bevor Sie ein Modell freigeben können, benötigen Sie einen trainierten benutzerdefinierten Classifier oder einen benutzerdefinierten Entity Recognizer in Amazon Comprehend in Ihrem AWS-Konto. Weitere Informationen zum Trainieren benutzerdefinierter Modelle finden Sie unter [Benutzerdefinierte Klassifizierung](#) oder [Erkennung benutzerdefinierter Entitäten](#).

## Erforderliche Berechtigungen

### IAM-Richtlinienanweisung

Bevor Sie einem benutzerdefinierten Modell eine ressourcenbasierte Richtlinie hinzufügen können, benötigen Sie Berechtigungen in AWS Identity and Access Management (IAM). Ihrem Benutzer, Ihrer Gruppe oder Ihrer Rolle muss eine Richtlinie angefügt sein, damit Sie Modellrichtlinien erstellen, abrufen und löschen können, wie im folgenden Beispiel gezeigt.

Example IAM-Richtlinie zur Verwaltung ressourcenbasierter Richtlinien für benutzerdefinierte Modelle

```
{
  "Effect": "Allow",
  "Action": [
    "comprehend:PutResourcePolicy",
    "comprehend>DeleteResourcePolicy",
    "comprehend:DescribeResourcePolicy"
  ],
  "Resource": "arn:aws:comprehend:us-west-2:111122223333:document-classifier/foo/
  version/*"
}
```

Informationen zum Erstellen einer IAM-Richtlinie finden Sie unter [Erstellen von IAM-Richtlinien](#) im IAM-Benutzerhandbuch. Informationen zum Anfügen einer IAM-Richtlinie finden Sie unter [Hinzufügen und Entfernen von IAM-Identitätsberechtigungen](#) im IAM-Benutzerhandbuch.

## AWS KMS Schlüsselrichtlinienanweisung

Wenn Sie ein verschlüsseltes Modell freigeben, müssen Sie möglicherweise Berechtigungen für hinzufügenAWS KMS. Diese Anforderung hängt von der Art des KMS-Schlüssels ab, den Sie zum Verschlüsseln des Modells in Amazon Comprehend verwenden.

Ein AWS-eigener Schlüssel ist im Besitz eines -AWSService und wird von diesem verwaltet. Wenn Sie eine verwendenAWS-eigener Schlüssel, müssen Sie keine Berechtigungen für hinzufügenAWS KMS, und Sie können diesen Abschnitt überspringen.

Ein vom Kunden verwalteter Schlüssel ist ein Schlüssel, den Sie in Ihrem erstellen, besitzen und verwaltenAWS-Konto. Wenn Sie einen vom Kunden verwalteten Schlüssel verwenden, müssen Sie Ihrer KMS-Schlüsselrichtlinie eine Anweisung hinzufügen.

Die Richtlinienanweisung autorisiert eine oder mehrere Entitäten (z. B. Benutzer oder Konten), die AWS KMS Operationen auszuführen, die zum Entschlüsseln des Modells erforderlich sind.

Sie verwenden Bedingungsschlüssel, um das Confused-Deputy-Problem zu vermeiden. Weitere Informationen finden Sie unter [the section called "Serviceübergreifende Confused-Deputy-Prävention"](#).

Verwenden Sie die folgenden Bedingungsschlüssel in der Richtlinie, um die Entitäten zu validieren, die auf Ihren KMS-Schlüssel zugreifen. Wenn ein Benutzer das Modell importiert, AWS KMS überprüft , ob der ARN der Quellmodellversion mit der Bedingung übereinstimmt. Wenn Sie keine Bedingung in die Richtlinie aufnehmen, können die angegebenen Prinzipale Ihren KMS-Schlüssel verwenden, um jede Modellversion zu entschlüsseln:

- [aws:SourceArn](#) – Verwenden Sie diesen Bedingungsschlüssel mit den `kms:Decrypt` Aktionen `kms:GenerateDataKey` und .
- [kms:EncryptionContext](#) – Verwenden Sie diesen Bedingungsschlüssel mit den `kms:CreateGrant` Aktionen `kms:Decrypt`, und `kms:GenerateDataKey`.

Im folgenden Beispiel autorisiert die Richtlinie AWS-Konto 444455556666 , Version 1 des angegebenen Classifier-Modells zu verwenden, das gehörtAWS-Konto111122223333.

Example KMS-Schlüsselrichtlinie für den Zugriff auf eine bestimmte Classifier-Modellversion

```
{  
  "Version": "2012-10-17",
```

```

"Statement": [
  {
    "Effect": "Allow",
    "Principal": {
      "AWS":
        "arn:aws:iam::444455556666:root"
    },
    "Action": [
      "kms:Decrypt",
      "kms:GenerateDataKey"
    ],
    "Resource": "*",
    "Condition": {
      "StringEquals": {
        "aws:SourceArn":
          "arn:aws:comprehend:us-west-2:111122223333:document-
classifier/classifierName/version/1"
      }
    }
  },
  {
    "Effect": "Allow",
    "Principal": {
      "AWS": "arn:aws:iam::444455556666:root"
    },
    "Action": "kms:CreateGrant",
    "Resource": "*",
    "Condition": {
      "StringEquals": {
        "kms:EncryptionContext:aws:comprehend:arn":
          "arn:aws:comprehend:us-west-2:111122223333:document-
classifier/classifierName/version/1"
      }
    }
  }
]
}

```

Die folgende Beispielrichtlinie autorisiert Benutzer ExampleUser von AWS-Konto 444455556666 und ExampleRole von AWS-Konto 123456789012 zum Zugriff auf diesen KMS-Schlüssel über den Amazon Comprehend-Service.

Example KMS-Schlüsselrichtlinie, um den Zugriff auf den Amazon Comprehend-Service zu ermöglichen (Alternative 1).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::444455556666:user/ExampleUser",
          "arn:aws:iam::123456789012:role/ExampleRole"
        ]
      },
      "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey"
      ],
      "Resource": "*",
      "Condition": {
        "StringLike": {
          "aws:SourceArn": "arn:aws:comprehend:*"
        }
      }
    },
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::444455556666:user/ExampleUser",
          "arn:aws:iam::123456789012:role/ExampleRole"
        ]
      },
      "Action": "kms:CreateGrant",
      "Resource": "*",
      "Condition": {
        "StringLike": {
          "kms:EncryptionContext:aws:comprehend:arn": "arn:aws:comprehend:*"
        }
      }
    }
  ]
}
```

```
}
```

Die folgende Beispielrichtlinie autorisiert AWS-Konto 444455556666 , über den Amazon Comprehend-Service auf diesen KMS-Schlüssel zuzugreifen, wobei eine alternative Syntax zum vorherigen Beispiel verwendet wird.

Example KMS-Schlüsselrichtlinie, um den Zugriff auf den Amazon Comprehend-Service zu ermöglichen (Alternative 2).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::444455556666:root"
      },
      "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey",
        "kms:CreateGrant"
      ],
      "Resource": "*",
      "Condition": {
        "StringLike": {
          "kms:EncryptionContext:aws:comprehend:arn": "arn:aws:comprehend:*"
        }
      }
    }
  ]
}
```

Weitere Informationen finden Sie unter [Schlüsselrichtlinien in AWS KMS](#) im Entwicklerhandbuch für AWS Key Management Service.

## Ressourcenbasierte Richtlinien für benutzerdefinierte Modelle

Bevor ein Amazon Comprehend-Benutzer in einem anderen ein benutzerdefiniertes Modell aus Ihrem AWS Konto importieren AWS-Konto kann, müssen Sie ihn dazu autorisieren. Um sie zu autorisieren, fügen Sie der Modellversion, die Sie freigeben möchten, eine ressourcenbasierte Richtlinie hinzu. Eine ressourcenbasierte Richtlinie ist eine IAM-Richtlinie, die Sie einer Ressource in anfügenAWS.

Wenn Sie eine Ressourcenrichtlinie an eine benutzerdefinierte Modellversion anfügen, autorisiert die Richtlinie Benutzer, Gruppen oder Rollen, die `comprehend:ImportModel` Aktion für die Modellversion auszuführen.

### Example Ressourcenbasierte Richtlinie für eine benutzerdefinierte Modellversion

In diesem Beispiel werden die autorisierten Entitäten im `Principal` Attribut angegeben. Ressource „\*“ bezieht sich auf die spezifische Modellversion, an die Sie die Richtlinie anfügen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "comprehend:ImportModel",
      "Resource": "*",
      "Principal": {
        "AWS": [
          "arn:aws:iam::111122223333:root",
          "arn:aws:iam::444455556666:user/ExampleUser",
          "arn:aws:iam::123456789012:role/ExampleRole"
        ]
      }
    }
  ]
}
```

Für Richtlinien, die Sie an benutzerdefinierte Modelle anfügen, `comprehend:ImportModel` ist die einzige Aktion, die Amazon Comprehend unterstützt.

Weitere Informationen zu ressourcenbasierten Richtlinien finden Sie unter [Identitätsbasierte Richtlinien und ressourcenbasierte Richtlinien](#) im IAM-Benutzerhandbuch.

### Schritt 1: Hinzufügen einer ressourcenbasierten Richtlinie zu einem benutzerdefinierten Modell

Sie können eine ressourcenbasierte Richtlinie mithilfe der AWS Management Console, AWS CLI oder Amazon Comprehend API hinzufügen.

#### AWS Management Console

Sie können Amazon Comprehend in der verwenden AWS Management Console.



## So fügen Sie eine ressourcenbasierte Richtlinie hinzu

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon Comprehend-Konsole unter <https://console.aws.amazon.com/comprehend/>
2. Wählen Sie im Navigationsmenü auf der linken Seite unter Anpassung die Seite aus, die Ihr benutzerdefiniertes Modell enthält:
  - a. Wenn Sie einen benutzerdefinierten Dokumentklassifizierer freigeben, wählen Sie Benutzerdefinierte Klassifizierung aus.
  - b. Wenn Sie eine benutzerdefinierte Entitätserkennung freigeben, wählen Sie Benutzerdefinierte Entitätserkennung aus.
3. Wählen Sie in der Liste der Modelle den Modellnamen aus, um die Detailseite zu öffnen.
4. Wählen Sie unter Versionen den Namen der Modellversion aus, die Sie teilen möchten.
5. Wählen Sie auf der Seite mit den Versionsdetails die Registerkarte Tags, VPC und Richtlinie aus.
6. Wählen Sie im Abschnitt Ressourcenbasierte Richtlinie die Option Bearbeiten aus.
7. Gehen Sie auf der Seite Ressourcenbasierte Richtlinie bearbeiten wie folgt vor:
  - a. Geben Sie unter Richtliniennamen einen Namen ein, der Ihnen hilft, die Richtlinie zu erkennen, nachdem Sie sie erstellt haben.
  - b. Geben Sie unter Autorisieren eine oder mehrere der folgenden Entitäten an, um sie zum Importieren Ihres Modells zu autorisieren:

Feld	Definition und Beispiele
Service-Prinzipal	Service-Prinzipal-IDs für die Services, die auf diese Modellversion zugreifen können. Beispielsweise:  comprehend.amazonaws.com
AWS-Konto IDs	AWS-Konten , die auf diese Modellversion zugreifen können. Autorisiert alle Benutzer, die zum Konto gehören. Beispielsweise:  111122223333, 123456789012

Feld	Definition und Beispiele
IAM-Entitäten	ARNs für Benutzer oder Rollen, die auf diese Modellversion zugreifen können. Beispielsweise:  arn:aws:iam::111122223333:user/ExampleUser, arn:aws:iam::444455556666:role/ExampleRole

8. Unter Freigeben können Sie den ARN der Modellversion kopieren, um sie mit der Person zu teilen, die Ihr Modell importieren wird. Wenn jemand ein benutzerdefiniertes Modell aus einem anderen importiertAWS-Konto, ist der ARN der Modellversion erforderlich.
9. Wählen Sie Speichern. Amazon Comprehend erstellt Ihre ressourcenbasierte Richtlinie und fügt sie Ihrem Modell hinzu.

## AWS CLI

Verwenden Sie den [PutResourcePolicy](#) Befehl AWS CLI, um einem benutzerdefinierten Modell mit der eine ressourcenbasierte Richtlinie hinzuzufügen. Der `-`Befehl verwendet die folgenden Parameter:

- `resource-arn` – Der ARN des benutzerdefinierten Modells, einschließlich der Modellversion.
- `resource-policy` – Eine JSON-Datei, die die ressourcenbasierte Richtlinie definiert, die an Ihr benutzerdefiniertes Modell angehängt werden soll.

Sie können die Richtlinie auch als Inline-JSON-Zeichenfolge angeben. Um gültiges JSON für Ihre Richtlinie anzugeben, schließen Sie die Attributnamen und Werte in doppelte Anführungszeichen ein. Wenn der JSON-Text auch in doppelte Anführungszeichen eingeschlossen ist, maskieren Sie die doppelten Anführungszeichen, die sich in der Richtlinie befinden.

- `policy-revision-id` – Die Revisions-ID, die Amazon Comprehend der Richtlinie zugewiesen hat, die Sie aktualisieren. Wenn Sie eine neue Richtlinie ohne vorherige Version erstellen, verwenden Sie diesen Parameter nicht. Amazon Comprehend erstellt die Revisions-ID für Sie.

Example Hinzufügen einer ressourcenbasierten Richtlinie zu einem benutzerdefinierten Modell mit dem **put-resource-policy** Befehl

In diesem Beispiel wird eine Richtlinie in einer JSON-Datei mit dem Namen `policyFile.json` definiert und die Richtlinie einem Modell zugeordnet. Das Modell ist Version v2 eines Classifiers namens `mycf1`.

```
$ aws comprehend put-resource-policy \  
> --resource-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier/mycf1/  
version/v2 \  
> --resource-policy file://policyFile.json \  
> --policy-revision-id revision-id
```

Die JSON-Datei für die Ressourcenrichtlinie enthält den folgenden Inhalt:

- Aktion – Die Richtlinie autorisiert die benannten Prinzipale zur Verwendung von `comprehend:ImportModel`.
- Ressource – Der ARN des benutzerdefinierten Modells. Ressource „\*“ bezieht sich auf die Modellversion, die Sie im `put-resource-policy` Befehl angeben.
- Prinzipal – Die Richtlinie autorisiert Benutzer jane aus AWS-Konto 444455556666 und alle Benutzer aus AWS-Konto 123456789012.

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {"Sid": "ResourcePolicyForImportModel",  
      "Effect": "Allow",  
      "Action": ["comprehend:ImportModel"],  
      "Resource": "*",  
      "Principal":  
        {"AWS":  
          ["arn:aws:iam::444455556666:user/jane",  
            "123456789012"]  
        }  
    }  
  ]  
}
```

## Amazon Comprehend API

Um einem benutzerdefinierten Modell mithilfe der Amazon Comprehend API eine ressourcenbasierte Richtlinie hinzuzufügen, verwenden Sie die [PutResourcePolicy](#) -API-Operation.

Sie können einem benutzerdefinierten Modell in der API-Anforderung, die das Modell erstellt, auch eine Richtlinie hinzufügen. Geben Sie dazu das Richtlinien-JSON für den ModelPolicy Parameter an, wenn Sie eine - [CreateDocumentClassifier](#) oder - [CreateEntityRecognizer](#) Anforderung senden.

## Schritt 2: Geben Sie die Details an, die andere importieren müssen

Nachdem Sie Ihrem benutzerdefinierten Modell die ressourcenbasierte Richtlinie hinzugefügt haben, haben Sie andere Amazon Comprehend-Benutzer autorisiert, Ihr Modell in ihr zu importieren AWS-Konten. Bevor sie jedoch importieren können, müssen Sie ihnen die folgenden Details zur Verfügung stellen:

- Der Amazon-Ressourcenname (ARN) der Modellversion.
- Die AWS-Region, die das Modell enthält. Jeder, der Ihr Modell importiert, muss denselben verwenden AWS-Region.
- Ob das Modell verschlüsselt ist und wenn ja, welcher AWS KMS Schlüsseltyp verwendet wird: AWS-eigener Schlüssel oder vom Kunden verwalteter Schlüssel.
- Wenn Ihr Modell mit einem vom Kunden verwalteten Schlüssel verschlüsselt ist, müssen Sie den ARN des KMS-Schlüssels angeben. Jeder, der Ihr Modell importiert, muss den ARN in eine IAM-Servicerolle in seine aufnehmen AWS-Konto. Diese Rolle autorisiert Amazon Comprehend, den KMS-Schlüssel zum Entschlüsseln des Modells während des Imports zu verwenden.

Weitere Informationen darüber, wie andere Benutzer Ihr Modell importieren, finden Sie unter [Importieren eines benutzerdefinierten Modells aus einem anderen AWS-Konto](#).

## Importieren eines benutzerdefinierten Modells aus einem anderen AWS-Konto

In Amazon Comprehend können Sie ein benutzerdefiniertes Modell importieren, das sich in einem anderen befindet AWS-Konto. Wenn Sie ein Modell importieren, erstellen Sie ein neues benutzerdefiniertes Modell in Ihrem Konto. Ihr neues benutzerdefiniertes Modell ist ein vollständig trainiertes Duplikat des Modells, das Sie importiert haben.

### Themen

- [Bevor Sie beginnen](#)
- [Importieren eines benutzerdefinierten Modells](#)

## Bevor Sie beginnen

Bevor Sie ein benutzerdefiniertes Modell aus einem anderen importieren können AWS-Konto, stellen Sie sicher, dass die Person, die das Modell für Sie freigegeben hat, Folgendes tut:

- Autorisiert Sie zum Importieren. Diese Autorisierung wird in der ressourcenbasierten Richtlinie gewährt, die an die Modellversion angehängt ist. Weitere Informationen finden Sie unter [Ressourcenbasierte Richtlinien für benutzerdefinierte Modelle](#).
- Stellt Ihnen die folgenden Informationen zur Verfügung:
  - Der Amazon-Ressourcenname (ARN) der Modellversion.
  - Die AWS-Region, die das Modell enthält. Sie müssen diese AWS-Region beim Import verwenden.
  - Gibt an, ob das Modell mit einem -AWS KMS-Schlüssel verschlüsselt ist, und, falls dies der Fall ist, den verwendeten Schlüsseltyp.

Wenn das Modell verschlüsselt ist, müssen Sie je nach Art des verwendeten KMS-Schlüssels möglicherweise zusätzliche Schritte unternehmen:

- AWS-eigener Schlüssel – Diese Art von KMS-Schlüssel gehört und wird von verwaltet AWS. Wenn das Modell mit einem verschlüsselt ist AWS-eigener Schlüssel, sind keine zusätzlichen Schritte erforderlich.
- Vom Kunden verwalteter Schlüssel – Diese Art von KMS-Schlüssel wird von einem -AWS Kunden in seinem erstellt, besessen und verwaltet AWS-Konto. Wenn das Modell mit einem vom Kunden verwalteten Schlüssel verschlüsselt ist, muss die Person, die das Modell freigegeben hat,:
  - Autorisieren Sie Sie zum Entschlüsseln des Modells. Diese Autorisierung wird in der KMS-Schlüsselrichtlinie für den vom Kunden verwalteten Schlüssel erteilt. Weitere Informationen finden Sie unter [AWS KMS Schlüsselrichtlinienanweisung](#).
  - Geben Sie den ARN des vom Kunden verwalteten Schlüssels an. Sie verwenden diesen ARN, wenn Sie eine IAM-Service-Rolle erstellen. Diese Rolle autorisiert Amazon Comprehend, den KMS-Schlüssel zum Entschlüsseln des Modells zu verwenden.

## Erforderliche Berechtigungen

Bevor Sie ein benutzerdefiniertes Modell importieren können, müssen Sie oder Ihr Administrator die erforderlichen Aktionen in AWS Identity and Access Management (IAM) autorisieren. Als Amazon Comprehend-Benutzer müssen Sie durch eine IAM-Richtlinienanweisung zum Importieren autorisiert

sein. Wenn während des Imports eine Verschlüsselung oder Entschlüsselung erforderlich ist, muss Amazon Comprehend zur Verwendung der erforderlichen AWS KMS Schlüssel autorisiert sein.

### IAM-Richtlinienanweisung

Ihrem Benutzer, Ihrer Gruppe oder Ihrer Rolle muss eine Richtlinie zugeordnet sein, die die `ImportModel` Aktion zulässt, wie im folgenden Beispiel gezeigt.

Example IAM-Richtlinie zum Importieren eines benutzerdefinierten Modells

```
{
  "Effect": "Allow",
  "Action": [
    "comprehend:ImportModel"
  ],
  "Resource": "arn:aws:comprehend:us-west-2:111122223333:document-classifier/foo/
version/*"
}
```

Informationen zum Erstellen einer IAM-Richtlinie finden Sie unter [Erstellen von IAM-Richtlinien](#) im IAM-Benutzerhandbuch. Informationen zum Anfügen einer IAM-Richtlinie finden Sie unter [Hinzufügen und Entfernen von IAM-Identitätsberechtigungen](#) im IAM-Benutzerhandbuch.

### IAM-Servicerolle für die AWS KMS Verschlüsselung

Wenn Sie ein benutzerdefiniertes Modell importieren, müssen Sie Amazon Comprehend autorisieren, AWS KMS Schlüssel in einem der folgenden Fälle zu verwenden:

- Sie importieren ein benutzerdefiniertes Modell, das mit einem vom Kunden verwalteten Schlüssel in verschlüsselt ist AWS KMS. In diesem Fall benötigt Amazon Comprehend Zugriff auf den KMS-Schlüssel, damit es das Modell während des Imports entschlüsseln kann.
- Sie möchten das neue benutzerdefinierte Modell, das Sie mit dem Import erstellen, verschlüsseln und einen vom Kunden verwalteten Schlüssel verwenden. In diesem Fall benötigt Amazon Comprehend Zugriff auf Ihren KMS-Schlüssel, damit er das neue Modell verschlüsseln kann.

Um Amazon Comprehend zur Verwendung dieser AWS KMS Schlüssel zu autorisieren, erstellen Sie eine IAM-Servicerolle. Diese Art von IAM-Rolle ermöglicht es einem -AWS-Service, in Ihrem Namen auf Ressourcen in anderen -Services zuzugreifen. Weitere Informationen zu Servicerollen finden Sie unter [Erstellen einer Rolle zum Delegieren von Berechtigungen an einen -AWS-Service](#) im IAM-Benutzerhandbuch.

Wenn Sie die Amazon Comprehend-Konsole zum Importieren verwenden, kann Amazon Comprehend die Servicerolle für Sie erstellen. Andernfalls müssen Sie eine Servicerolle in IAM erstellen, bevor Sie importieren.

Die IAM-Servicerolle muss über eine Berechtigungsrichtlinie und eine Vertrauensrichtlinie verfügen, wie in den folgenden Beispielen gezeigt.

### Example Berechtigungsrichtlinie

Die folgende Berechtigungsrichtlinie erlaubt die AWS KMS Operationen, die Amazon Comprehend zum Verschlüsseln und Entschlüsseln benutzerdefinierter Modelle verwendet. Sie gewährt Zugriff auf zwei KMS-Schlüssel:

- Ein KMS-Schlüssel befindet sich in der AWS-Konto, die das zu importierende Modell enthält. Es wurde verwendet, um das Modell zu verschlüsseln, und Amazon Comprehend verwendet es, um das Modell während des Imports zu entschlüsseln.
- Der andere KMS-Schlüssel befindet sich in der AWS-Konto, die das Modell importiert. Amazon Comprehend verwendet diesen Schlüssel, um das neue benutzerdefinierte Modell zu verschlüsseln, das durch den Import erstellt wird.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:CreateGrant"
      ],
      "Resource": [
        "arn:aws:kms:us-west-2:111122223333:key/key-id",
        "arn:aws:kms:us-west-2:444455556666:key/key-id"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt",
        "kms:GenerateDatakey"
      ],
      "Resource": [
        "arn:aws:kms:us-west-2:111122223333:key/key-id",
```

```

        "arn:aws:kms:us-west-2:444455556666:key/key-id"
    ],
    "Condition": {
        "StringEquals": {
            "kms:ViaService": [
                "s3.us-west-2.amazonaws.com"
            ]
        }
    }
}
]
}

```

## Example Vertrauensrichtlinie

Die folgende Vertrauensrichtlinie ermöglicht es Amazon Comprehend, die Rolle zu übernehmen und ihre Berechtigungen zu erhalten. Es ermöglicht dem `comprehend.amazonaws.com` Service-Prinzipal, den `sts:AssumeRole` Vorgang auszuführen. Um bei der [Prävention von verwirrem Stellvertreter](#) zu helfen, schränken Sie den Umfang der Berechtigung ein, indem Sie einen oder mehrere globale Bedingungskontextschlüssel verwenden. `aws:SourceAccount` Geben Sie für die Konto-ID des Benutzers an, der das Modell importiert.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "comprehend.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "444455556666"
        }
      }
    }
  ]
}

```



## Importieren eines benutzerdefinierten Modells

Sie können ein benutzerdefiniertes Modell mithilfe der AWS Management Console, AWS CLI oder Amazon Comprehend API importieren.

### AWS Management Console

Sie können Amazon Comprehend in der verwenden AWS Management Console.

So importieren Sie ein benutzerdefiniertes Modell

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon Comprehend-Konsole unter <https://console.aws.amazon.com/comprehend/>
2. Wählen Sie im Navigationsmenü auf der linken Seite unter Anpassung die Seite für den Modelltyp aus, den Sie importieren:
  - a. Wenn Sie einen benutzerdefinierten Dokumentklassifizierer importieren, wählen Sie Benutzerdefinierte Klassifizierung aus.
  - b. Wenn Sie eine benutzerdefinierte Entitätserkennung importieren, wählen Sie Benutzerdefinierte Entitätserkennung aus.
3. Wählen Sie Version importieren aus.
4. Geben Sie auf der Seite Modellversion importieren die folgenden Details ein:
  - ARN der Modellversion – Der ARN der zu importierenden Modellversion.
  - Modellname – Ein benutzerdefinierter Name für das neue Modell, das durch den Import erstellt wird.
  - Versionsname – Ein benutzerdefinierter Name für die neue Modellversion, die durch den Import erstellt wird.
5. Wählen Sie für Modellverschlüsselung den Typ des KMS-Schlüssels aus, der zum Verschlüsseln des neuen benutzerdefinierten Modells verwendet werden soll, das Sie mit dem Import erstellen:
  - Schlüssel im AWS Besitz von verwenden – Amazon Comprehend verschlüsselt Ihr Modell mit einem Schlüssel in AWS Key Management Service (AWS KMS), der von in Ihrem Namen erstellt, verwaltet und verwendet wird AWS.
  - Auswählen eines anderen AWS KMS Schlüssels (erweitert) – Amazon Comprehend verschlüsselt Ihr Modell mithilfe eines vom Kunden verwalteten Schlüssels, den Sie in verwalten AWS KMS.

Wenn Sie diese Option wählen, wählen Sie einen KMS-Schlüssel aus, der sich in Ihrem AWS-Konto befindet, oder erstellen Sie einen neuen, indem Sie [AWS KMS Schlüssel erstellen](#) auswählen.

6. Gewähren Sie Amazon Comprehend im Abschnitt **Servicezugriff** Zugriff auf alle AWS KMS Schlüssel, die es für Folgendes benötigt:
  - Entschlüsseln Sie das benutzerdefinierte Modell, das Sie importieren.
  - Verschlüsseln Sie das neue benutzerdefinierte Modell, das Sie mit dem Import erstellen.

Sie gewähren Zugriff mit einer IAM-Servicerolle, die es Amazon Comprehend ermöglicht, die KMS-Schlüssel zu verwenden.

Führen Sie für Servicerolle einen der folgenden Schritte aus:

- Wenn Sie über eine vorhandene Servicerolle verfügen, die Sie verwenden möchten, wählen Sie **Vorhandene IAM-Rolle verwenden** aus. Wählen Sie sie dann unter **Rollename** aus.
  - Wenn Amazon Comprehend eine Rolle für Sie erstellen soll, wählen Sie **Erstellen einer IAM-Rolle** aus.
7. Wenn Sie Amazon Comprehend die Rolle für Sie erstellen lassen möchten, gehen Sie wie folgt vor:
    - a. Geben Sie für **Rollename** ein Suffix für den Rollennamen ein, das Ihnen später hilft, die Rolle zu erkennen.
    - b. Geben Sie für **Quell-KMS-Schlüssel-ARN** den ARN des KMS-Schlüssels ein, der zum Verschlüsseln des Modells verwendet wird, das Sie importieren. Amazon Comprehend verwendet diesen Schlüssel, um das Modell während des Imports zu entschlüsseln.
  8. (Optional) Im Abschnitt **Tags** können Sie dem neuen benutzerdefinierten Modell, das Sie durch Importieren erstellen, Tags hinzufügen. Weitere Informationen zum Markieren von benutzerdefinierten Modellen finden Sie unter [Eine neue Ressource taggen](#).
  9. Wählen Sie **Bestätigen** aus.

## AWS CLI

Sie können Amazon Comprehend verwenden, indem Sie Befehle mit der `aws` CLI ausführen.

## Example Import-Model-Befehl

Verwenden Sie den `import-model` Befehl , um ein benutzerdefiniertes Modell zu importieren:

```
$ aws comprehend import-model \  
> --source-model arn:aws:comprehend:us-west-2:111122223333:document-classifier/foo/  
version/bar \  
> --model-name importedDocumentClassifier \  
> --version-name versionOne \  
> --data-access-role-arn arn:aws:iam::444455556666:role/comprehendAccessRole \  
> --model-kms-key-id kms-key-id
```

In diesem Beispiel werden die folgenden Parameter verwendet:

- `source-model` – Der ARN des zu importierenden benutzerdefinierten Modells.
- `model-name` – Ein benutzerdefinierter Name für das neue Modell, das durch den Import erstellt wird.
- `version-name` – Ein benutzerdefinierter Name für die neue Modellversion, die durch den Import erstellt wird.
- `data-access-role-arn` – Der ARN der IAM-Service-Rolle, die es Amazon Comprehend ermöglicht, die erforderlichen AWS KMS Schlüssel zum Verschlüsseln oder Entschlüsseln des benutzerdefinierten Modells zu verwenden.
- `model-kms-key-id` – Der ARN oder die ID des KMS-Schlüssels, den Amazon Comprehend zum Verschlüsseln des benutzerdefinierten Modells verwendet, das Sie mit diesem Import erstellen. Dieser Schlüssel muss sich in AWS KMS Ihres AWS-Konto befinden.

## Amazon Comprehend API

Um ein benutzerdefiniertes Modell mithilfe der Amazon Comprehend API zu importieren, verwenden Sie die [ImportModel](#) -API-Aktion.

# Flywheels

Ein Amazon Comprehend Flywheel vereinfacht die Verbesserung eines benutzerdefinierten Modells im Laufe der Zeit. Sie können ein Flywheel verwenden, um die Aufgaben im Zusammenhang mit dem Training und der Auswertung neuer benutzerdefinierter Modellversionen zu orchestrieren. Flywheels unterstützen benutzerdefinierte Klartextmodelle für die benutzerdefinierte Klassifizierung und die Erkennung benutzerdefinierter Entitäten.

## Themen

- [Überblick über das Flywheel](#)
- [Flywheel-Data-Lakes](#)
- [IAM-Richtlinien und -Berechtigungen](#)
- [Konfigurieren von Flywheels mit der Konsole](#)
- [Konfigurieren von Flywheels mit der API](#)
- [Konfigurieren von Datensätzen](#)
- [Flywheel-Iterationen](#)
- [Verwenden von Flywheels für Analysen](#)

## Überblick über das Flywheel

Ein Flywheel ist eine Amazon Comprehend-Ressource, die das Training und die Auswertung neuer Versionen eines benutzerdefinierten Modells orchestriert. Sie können ein Flywheel erstellen, um ein vorhandenes trainiertes Modell zu verwenden, oder Amazon Comprehend kann ein neues Modell für das Flywheel erstellen und trainieren. Verwenden Sie Flywheels mit benutzerdefinierten Klartextmodellen für die benutzerdefinierte Klassifizierung oder die Erkennung benutzerdefinierter Entitäten.

Sie können Flywheels mithilfe der Amazon Comprehend-Konsole oder der API konfigurieren und verwalten. Sie können Flywheels auch mit konfigurierenAWS CloudFormation.

Wenn Sie ein Flywheel erstellen, erstellt Amazon Comprehend einen Data Lake in Ihrem Konto. Der [Data Lake](#) speichert und verwaltet alle Flywheel-Daten, z. B. die Trainingsdaten und Testdaten für alle Versionen des Modells.

Sie legen die aktive Modellversion als Version des Flywheel-Modells fest, das Sie für Inferenzaufträge oder Amazon Comprehend-Endpunkte verwenden möchten. Anfänglich enthält das Flywheel

eine Version des Modells. Im Laufe der Zeit wählen Sie beim Trainieren neuer Modellversionen die leistungsstärkste Version aus, die die aktive Modellversion sein soll. Wenn ein Benutzer den Flywheel-ARN angibt, um einen Inferenzauftrag auszuführen, führt Amazon Comprehend den Auftrag mit der aktiven Modellversion des Flywheels aus.

In regelmäßigen Abständen erhalten Sie neue beschriftete Daten (Trainingsdaten oder Testdaten) für das Modell. Sie stellen dem Flywheel neue Daten zur Verfügung, indem Sie einen oder mehrere Datensätze erstellen. Ein Datensatz enthält Eingabedaten für das Training oder Testen des benutzerdefinierten Modells, das einem Flywheel zugeordnet ist. Amazon Comprehend lädt die Eingabedaten in den Data Lake des Flywheel hoch.

Um die neuen Datensätze in Ihr benutzerdefiniertes Modell zu integrieren, erstellen und führen Sie eine Flywheel-Iteration aus. Eine Flywheel-Iteration ist ein Workflow, der die neuen Datensätze verwendet, um die aktive Modellversion auszuwerten und eine neue Modellversion zu trainieren. Basierend auf den Metriken für die vorhandenen und neuen Modellversionen können Sie entscheiden, ob die neue Modellversion zur aktiven Version hochgestuft werden soll.

Sie können die aktive Modellversion des Flywheels verwenden, um benutzerdefinierte Analysen auszuführen (in Echtzeit oder asynchrone Aufträge). Um das Flywheel-Modell für die Echtzeitanalyse zu verwenden, müssen Sie einen [Endpunkt](#) für das Flywheel erstellen.

Für die Nutzung von Flywheels fallen keine zusätzlichen Gebühren an. Wenn Sie jedoch eine Flywheel-Iteration ausführen, fallen die Standardgebühren für das Training einer neuen Modellversion und das Speichern der Modelldaten an. Ausführliche Preisinformationen finden Sie unter [Amazon Comprehend – Preise](#).

## Themen

- [Flywheel-Datensätze](#)
- [Flywheel-Erstellung](#)
- [Flywheel-Zustände](#)
- [Flywheel-Iterationen](#)

## Flywheel-Datensätze

Um einem Flywheel neue beschriftete Daten hinzuzufügen, erstellen Sie einen Datensatz. Sie konfigurieren jeden Datensatz als Trainings- oder Testdaten. Sie verknüpfen den Datensatz mit einem bestimmten Flywheel und einem benutzerdefinierten Modell.

Nachdem Sie einen Datensatz erstellt haben, lädt Amazon Comprehend die Daten in den Data Lake des Flywheel hoch. Weitere Informationen finden Sie unter [Flywheel-Data-Lakes](#).

## Flywheel-Erstellung

Wenn Sie ein Flywheel erstellen, können Sie das Flywheel einem vorhandenen trainierten Modell zuordnen oder das Flywheel kann ein neues Modell erstellen.

Wenn Sie ein Flywheel mit einem vorhandenen Modell erstellen, geben Sie die aktive Modellversion an. Amazon Comprehend kopiert die Trainings- und Testdaten des Modells in den Data Lake des Flywheels. Stellen Sie sicher, dass die Modelltrainings- und Testdaten am selben Amazon S3-Speicherort vorhanden sind wie beim Erstellen des Modells.

Um ein Flywheel für ein neues Modell zu erstellen, stellen Sie einen Datensatz für Trainingsdaten (und einen optionalen Datensatz für Testdaten) bereit, wenn Sie das Flywheel erstellen. Wenn Sie das Flywheel ausführen, um die erste Flywheel-Iteration zu erstellen, trainiert das Flywheel das neue Modell.

Wenn Sie ein benutzerdefiniertes Modell trainieren, geben Sie eine Liste von benutzerdefinierten Labels (benutzerdefinierte Klassifizierung) oder benutzerdefinierten Entitäten (benutzerdefinierte Entitätserkennung) an, die das Modell erkennen soll. Beachten Sie die folgenden wichtigen Punkte zu benutzerdefinierten Labels/Entitäten:

- Wenn Sie ein Flywheel für ein neues Modell erstellen, ist die Liste der Labels/Entitäten, die Sie bei der Erstellung des Flywheels angeben, die endgültige Liste für das Flywheel.
- Wenn Sie ein Flywheel aus einem vorhandenen Modell erstellen, wird die Liste der diesem Modell zugeordneten Labels/Entitäten zur endgültigen Liste für das Flywheel.
- Wenn Sie dem Flywheel einen neuen Datensatz zuordnen und dieser Datensatz zusätzliche Beschriftungen/Entitäten enthält, ignoriert Amazon Comprehend die neuen Beschriftungen/Entitäten.
- Sie können die Bezeichnungs-/Entitätsliste eines Flywheels mithilfe der [DescribeFlywheel](#) API-Operation überprüfen.

### Note

Für die benutzerdefinierte Klassifizierung füllt Amazon Comprehend die Labelliste aus, nachdem der Flywheel-Status ACTIVE lautet. Warten Sie, bis das Flywheel aktiv ist, bevor Sie die DescribeFlywheel API-Operation aufrufen.

## Flywheel-Zustände

Ein Flywheel wechselt zwischen den folgenden Zuständen:

- **ERSTELLEN** – Amazon Comprehend erstellt die Flywheel-Ressourcen. Sie können Lesevorgänge am Flywheel ausführen, z. B. `DescribeFlywheel`.
- **ACTIVE** – Das Flywheel ist aktiv. Sie können feststellen, ob eine Flywheel-Iteration läuft, und den Status der Iteration anzeigen. Sie können Leseaktionen für das Flywheel und Aktionen wie `DeleteFlywheel` und `ausführenUpdateFlywheel`.
- **UPDATING** – Amazon Comprehend aktualisiert das Flywheel. Sie können Lesevorgänge für das Flywheel ausführen.
- **DELETING** – Amazon Comprehend löscht das Flywheel. Sie können Lesevorgänge für das Flywheel ausführen.
- **FAILED** – Die Erstellung des Flywheel ist fehlgeschlagen.

Nachdem Amazon Comprehend ein Flywheel gelöscht hat, behalten Sie den Zugriff auf alle Modelldaten im Flywheel Data Lake. Amazon Comprehend löscht alle internen Metadaten, die für die Verwaltung der Flywheel-Ressourcen erforderlich sind. Amazon Comprehend löscht auch die mit diesem Flywheel verknüpften Datensätze (die Modelldaten werden im Data Lake gespeichert).

## Flywheel-Iterationen

Wenn Sie neue Trainings- oder Testdaten für ein Flywheel-Modell erhalten, erstellen Sie einen oder mehrere neue Datensätze, um die neuen Daten in den Data Lake des Flywheels hochzuladen.

Anschließend führen Sie das Flywheel aus, um eine neue Flywheel-Iteration zu erstellen. Die Flywheel-Iteration wertet die aktuelle aktive Modellversion anhand der neuen Daten aus und speichert die Ergebnisse im Data Lake. Das Flywheel erstellt und trainiert auch eine neue Modellversion.

Wenn das neue Modell eine bessere Leistung als die aktuelle aktive Modellversion aufweist, können Sie die neue Modellversion zur aktiven Modellversion hochstufen. Sie können die [-Konsole](#) oder die [UpdateFlywheel](#) -API-Operation verwenden, um die aktive Modellversion zu aktualisieren.

# Flywheel-Data-Lakes

Wenn Sie ein Flywheel erstellen, erstellt Amazon Comprehend einen Data Lake in Ihrem Konto, der alle Flywheel-Daten enthält, z. B. die für die Modellversionen erforderlichen Eingabe- und Ausgabedaten.

Amazon Comprehend erstellt den Data Lake an dem Amazon S3-Speicherort, den Sie beim Erstellen des Flywheels angeben. Sie können den Speicherort als Amazon S3-Bucket oder als neuen Ordner in einem Amazon S3-Bucket angeben.

## Data-Lake-Ordnerstruktur

Wenn Amazon Comprehend den Data Lake erstellt, richtet es die folgende Ordnerstruktur am Amazon S3-Speicherort ein.

### Warning

Amazon Comprehend verwaltet die Organisation und den Inhalt des Data-Lake-Ordners. Verwenden Sie immer die Amazon Comprehend API-Operationen, um die Data Lake-Ordner zu ändern, da sonst Ihr Flywheel möglicherweise nicht ordnungsgemäß funktioniert.

```
Document Pool
Annotations Pool
Staging
Model Datasets
  (data for each version of the model)
  VersionID-1
    Training
    Test
    ModelStats
  VersionID-2
    Training
    Test
    ModelStats
```

Führen Sie die folgenden Schritte aus, um die Trainingsbewertung einer Modellversion anzuzeigen:

1. Öffnen Sie den Ordner mit dem Namen Model Datasets auf der Stammebene des Data Lake. Dieser Ordner enthält einen Unterordner für jede Version des Modells.



2. Öffnen Sie den Ordner für die gewünschte Modellversion.
3. Öffnen Sie den Ordner mit dem Namen ModelStats, um die Statistiken für das Modell anzuzeigen.

## Data-Lake-Management

Amazon Comprehend führt die folgenden Aufgaben aus, um den Data Lake in Ihrem Namen zu verwalten:

- Definiert die Ordnerstruktur des Data Lake und nimmt Datensätze in die entsprechenden Ordner auf.
- Verwaltet die Eingabedokumente (wie Textdateien und Anmerkungsdateien), die zum Trainieren des Modells erforderlich sind.
- Verwaltet die Trainings- und Bewertungsausgabedaten, die jeder Version des Modells zugeordnet sind.
- Verwaltet die Verschlüsselung für Dateien, die im Data Lake gespeichert sind.

Amazon Comprehend führt alle Datenerstellungs- und Aktualisierungsvorgänge für den Data Lake durch. Sie behalten vollen Zugriff auf die Daten im Data Lake. Beispielsweise:

- Sie haben vollen Zugriff auf den Inhalt des Data Lake.
- Der Data Lake bleibt verfügbar, nachdem Sie das Flywheel gelöscht haben.
- Sie können Zugriffsprotokolle für den Amazon S3-Bucket konfigurieren, der den Data Lake enthält.
- Sie können Verschlüsselungsschlüssel für die Daten bereitstellen. Sie geben diese beim Erstellen des Flywheel an.

Wir empfehlen Ihnen, die folgenden bewährten Methoden:

- Fügen Sie Ihre eigenen Ordner oder Dateien nicht manuell zum Data Lake hinzu. Ändern oder löschen Sie keine Dateien im Data Lake.
- Verwenden Sie immer die Erstellungs- und Aktualisierungsvorgänge von Amazon Comprehend, um Daten im Data Lake hinzuzufügen oder zu ändern. Verwenden Sie beispielsweise `CreateDataset` um Trainings- oder Testdaten bereitzustellen und Bewertungsdaten für Modellversionen `StartFlywheelIteration` zu generieren.

- Die Data-Lake-Struktur kann sich im Laufe der Zeit weiterentwickeln. Erstellen Sie keine Downstream-Skripte oder Programme, die explizit auf der Data-Lake-Struktur basieren.
- Wenn Sie einen Data-Lake-Speicherort für das Flywheel angeben, empfehlen wir, ein gemeinsames Präfix für Daten zu erstellen, die sich auf alle Flywheels beziehen, oder für jedes Flywheel ein anderes Präfix zu verwenden. Wir empfehlen nicht, den vollständigen Data-Lake-Pfad eines Flywheels als Präfix für ein anderes Flywheel zu verwenden.

## IAM-Richtlinien und -Berechtigungen

Sie konfigurieren die folgenden Richtlinien und Berechtigungen für die Verwendung von Flywheels:

- [the section called “Konfigurieren von IAM-Benutzerberechtigungen”](#) für Benutzer für den Zugriff auf Flywheel-Operationen.
- (Optional) [the section called “Konfigurieren von Berechtigungen für -AWS KMSSchlüssel”](#) für den Data Lake.
- [the section called “Erstellen einer Datenzugriffsrolle”](#) autorisiert Amazon Comprehend zum Zugriff auf den Data Lake.

## Konfigurieren von IAM-Benutzerberechtigungen

Um Flywheel-Funktionen zu verwenden, fügen Sie Ihren AWS Identity and Access Management (IAM)-Identitäten (Benutzer, Gruppen und Rollen) entsprechende Berechtigungsrichtlinien hinzu.

Das folgende Beispiel zeigt eine Berechtigungsrichtlinie zum Erstellen von Datensätzen, zum Erstellen und Verwalten von Flywheels und zum Ausführen des Flywheel.

Example IAM-Richtlinie zur Verwaltung von Flywheels

```
{
  "Effect": "Allow",
  "Action": [
    "comprehend:CreateFlywheel",
    "comprehend>DeleteFlywheel",
    "comprehend:UpdateFlywheel",
    "comprehend:ListFlywheels",
    "comprehend:DescribeFlywheel",
    "comprehend:CreateDataset",
    "comprehend:DescribeDataset",
```

```
"comprehend:ListDatasets",
"comprehend:StartFlywheelIteration",
"comprehend:DescribeFlywheelIteration",
"comprehend:ListFlywheelIterationHistory"
],
"Resource": "*"
}
```

Informationen zum Erstellen von IAM-Richtlinien für Amazon Comprehend finden Sie unter [So funktioniert Amazon Comprehend mit IAM](#).

## Konfigurieren von Berechtigungen für -AWS KMSSchlüssel

Wenn Sie AWS KMS Schlüssel für Ihre Daten im Data Lake verwenden, richten Sie die erforderlichen Berechtigungen ein. Weitere Informationen finden Sie unter [Erforderliche Berechtigungen für die Verwendung der KMS-Verschlüsselung](#).

## Erstellen einer Datenzugriffsrolle

Sie erstellen eine Datenzugriffsrolle in IAM für Amazon Comprehend, um auf Flywheel-Daten im Data Lake zuzugreifen. Wenn Sie die Konsole verwenden, um ein Flywheel zu erstellen, kann das System optional eine neue Rolle für diesen Zweck erstellen. Weitere Informationen finden Sie unter [Rollenbasierte Berechtigungen, die für asynchrone Operationen erforderlich sind](#).

## Konfigurieren von Flywheels mit der Konsole

Sie können die Amazon Comprehend-Konsole verwenden, um Flywheels zu erstellen, zu aktualisieren und zu löschen.

Wenn Sie ein Flywheel erstellen, erstellt Amazon Comprehend einen Data Lake, der alle Daten enthält, die das Flywheel benötigt, z. B. die Trainings- und Testdaten für jede Version des Modells.

Wenn Sie ein Flywheel löschen, löscht Amazon Comprehend weder den Data Lake noch das Modell, das dem Flywheel zugeordnet ist.

Lesen Sie die Informationen im Abschnitt [Flywheel-Erstellung](#) bevor Sie ein neues Flywheel erstellen.

### Themen

- [Erstellen eines Flywheel](#)
- [Aktualisieren eines Flywheel](#)
- [Löschen eines Flywheel](#)

## Erstellen eines Flywheel

Wenn Sie ein Flywheel erstellen, hängen die erforderlichen Konfigurationsfelder davon ab, ob das Flywheel für ein vorhandenes benutzerdefiniertes Modell oder ein neues Modell bestimmt ist.

So erstellen Sie ein Flywheel

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die [Amazon Comprehend-Konsole](#).
2. Wählen Sie im linken Menü Flywheels aus.
3. Wählen Sie in der Tabelle Flywheels die Option Neues Flywheel erstellen aus.
4. Geben Sie unter Flywheel-Name einen Namen für das Flywheel ein.
5. (Optional) Um ein Flywheel für ein vorhandenes Modell zu erstellen, konfigurieren Sie die Felder unter Aktive Modellversion .
  - a. Wählen Sie aus der Dropdown-Liste Modell ein Modell aus
  - b. Wählen Sie in der Dropdown-Liste Version die Modellversion aus.
6. (Optional) Um ein neues Classifier-Modell für das Flywheel zu erstellen, wählen Sie unter Benutzerdefinierter Modelltyp eine benutzerdefinierte Klassifizierung aus und konfigurieren Sie die Parameter in den folgenden Schritten.
  - a. Wählen Sie unter Sprache die Sprache für das Modell aus.
  - b. Wählen Sie unter Classifier-Modus die Option Single-Label-Modus oder Multi-Label-Modus aus.
  - c. Geben Sie unter Benutzerdefinierte Labels eine oder mehrere benutzerdefinierte Labels ein, die für das Training des Modells verwendet werden sollen. Jedes Label muss mit einer der Klassen in Ihren Eingabe-Trainingsdaten übereinstimmen.
7. (Optional) Um ein neues Entitätserkennungsmodell für das Flywheel zu erstellen, wählen Sie unter Benutzerdefinierter Modelltyp eine Benutzerdefinierte Entitätserkennung aus und konfigurieren Sie die Parameter in den folgenden Schritten.
  - a. Wählen Sie unter Sprache die Sprache für das Modell aus.

- b. Geben Sie unter Benutzerdefinierter Entitätstyp bis zu 25 benutzerdefinierte Entitäten ein, die für das Training des Modells verwendet werden sollen. Jedes Label muss mit einem der Entitätstypen in Ihren Eingabetrainingsdaten übereinstimmen.

Um mehr als eine Bezeichnung zu erstellen, führen Sie die folgenden Schritte mehrmals aus.

- i. Geben Sie eine benutzerdefinierte Bezeichnung ein. Die Bezeichnung muss ausschließlich Großbuchstaben enthalten. Verwenden Sie einen Unterstrich als Trennzeichen zwischen Wörtern in der Bezeichnung.
- ii. Wählen Sie Typ hinzufügen aus.

Um eine der hinzugefügten Beschriftungen zu entfernen, wählen Sie X rechts neben dem Beschriftungsnamen.

8. Konfigurieren Sie Ihre Wahl für Volume-Verschlüsselung, Modellverschlüsselung und Data-Lake-Verschlüsselung. Wählen Sie für jede dieser Optionen aus, ob Sie einen -AWS-eigenen KMS-Schlüssel oder einen Schlüssel verwenden möchten, zu dessen Verwendung Sie berechtigt sind.
  - Wenn Sie einen AWS-eigenen KMS-Schlüssel verwenden, gibt es keine zusätzlichen Parameter.
  - Wenn Sie einen anderen vorhandenen Schlüssel verwenden, geben Sie für KMS-Schlüssel-ARN den ARN für die Schlüssel-ID ein.
  - Wenn Sie einen neuen Schlüssel erstellen möchten, wählen Sie AWS KMS-Schlüssel erstellen aus.

Weitere Informationen zum Erstellen und Verwenden von KMS-Schlüsseln und der zugehörigen Verschlüsselung finden Sie unter [AWS Key Management Service](#).

- a. Konfigurieren Sie den Volume-Verschlüsselungsschlüssel. Amazon Comprehend verwendet diesen Schlüssel, um die Daten im Speicher-Volume zu verschlüsseln, während Ihr Auftrag verarbeitet wird. Wählen Sie aus, ob Sie einen AWS-eigenen KMS-Schlüssel oder einen Schlüssel verwenden möchten, zu dessen Verwendung Sie berechtigt sind.
  - b. Konfigurieren Sie den Modellverschlüsselungsschlüssel. Amazon Comprehend verwendet diesen Schlüssel, um die Modelldaten für diese Modellversion zu verschlüsseln.
9. Konfigurieren Sie den Data-Lake-Standort . Weitere Informationen finden Sie unter [Data-Lake-Management](#).

10. (Optional) Konfigurieren Sie den Data-Lake-Verschlüsselungsschlüssel. Amazon Comprehend verwendet diesen Schlüssel, um alle Dateien im Data Lake zu verschlüsseln.
11. (Optional) Konfigurieren Sie die VPC-Einstellungen . Geben Sie die VPC-ID unter VPC ein oder wählen Sie die ID aus der Dropdown-Liste aus.
  1. Wählen Sie das Subnetz unter Subnetz(e) aus. Nachdem Sie das erste Subnetz ausgewählt haben, können Sie weitere auswählen.
  2. Wählen Sie unter Sicherheitsgruppe(n) die zu verwendende Sicherheitsgruppe aus, wenn Sie eine angegeben haben. Nachdem Sie die erste Sicherheitsgruppe ausgewählt haben, können Sie zusätzliche auswählen.
12. Konfigurieren Sie die Service-Zugriffsberechtigungen.
  1. Wenn Sie Vorhandene IAM-Rolle verwenden auswählen, wählen Sie den Rollennamen in der Dropdown-Liste aus.
  2. Wenn Sie IAM-Rolle erstellen auswählen, erstellt Amazon Comprehend eine neue Rolle. Die Konsole zeigt die Berechtigungen an, die Amazon Comprehend für die Rolle konfiguriert. Geben Sie unter Rollenname einen beschreibenden Namen für die Rolle ein.
13. (Optional) Konfigurieren Sie Tags-Einstellungen. Um ein Tag hinzuzufügen, geben Sie unter Tags ein Schlüssel-Wert-Paar ein. Wählen Sie Add tag. Um dieses Paar zu entfernen, bevor Sie das Flywheel erstellen, wählen Sie Tag entfernen aus. Weitere Informationen finden Sie unter [Markieren Ihrer -Ressourcen](#).
14. Wählen Sie Erstellen.

## Aktualisieren eines Flywheel

Sie können den Flywheel-Namen, den Data-Lake-Standort, den Modelltyp und die Modellkonfiguration nur konfigurieren, wenn Sie das Flywheel erstellen.

Wenn Sie ein Flywheel aktualisieren, können Sie ein anderes Modell angeben, wenn der Modelltyp und die Konfigurationsoptionen mit dem aktuellen Modell übereinstimmen. Sie können eine neue aktive Modellversion konfigurieren. Sie können auch Verschlüsselungsdetails, Servicezugriffsberechtigungen und VPC-Einstellungen aktualisieren.

So aktualisieren Sie ein Flywheel

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die [Amazon Comprehend-Konsole](#) .

2. Wählen Sie im linken Menü Flywheels aus.
3. Wählen Sie in der Tabelle Flywheels das zu aktualisierende Flywheel aus.
4. Wählen Sie unter Aktive Modellversion ein Modell aus der Dropdown-Liste Modell und wählen Sie eine Modellversion aus.

Das Formular füllt den Modelltyp und die Modellkonfiguration aus.

5. (Optional) Konfigurieren Sie Volume-Verschlüsselungs- und Modellverschlüsselungseinstellungen.
6. (Optional) Konfigurieren Sie die Data-Lake-Verschlüsselungseinstellungen.
7. Konfigurieren Sie die Service-Zugriffsberechtigungen.
8. (Optional) Konfigurieren Sie die VPC-Einstellungen .
9. (Optional) Konfigurieren Sie Tags-Einstellungen.
10. Wählen Sie Speichern.

## Löschen eines Flywheel

So löschen Sie ein Flywheel

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die [Amazon Comprehend-Konsole](#) .
2. Wählen Sie im linken Menü Flywheels aus.
3. Wählen Sie in der Tabelle Flywheels das zu löschende Flywheel aus.
4. Wählen Sie Löschen.

## Konfigurieren von Flywheels mit der API

Sie können die Amazon Comprehend API verwenden, um Flywheels zu erstellen, zu aktualisieren und zu löschen.

Wenn Sie ein Flywheel erstellen, erstellt Amazon Comprehend einen Data Lake, der alle Daten enthält, die das Flywheel benötigt, z. B. die Trainings- und Testdaten für jede Version des Modells.

Wenn Sie ein Flywheel löschen, löscht Amazon Comprehend weder den Data Lake noch das mit dem Flywheel verknüpfte Modell.

Der Löschvorgang für Flywheel schlägt fehl, wenn das Flywheel eine Iteration ausführt oder einen Datensatz erstellt.

Lesen Sie die Informationen im Abschnitt , [Flywheel-Erstellung](#) bevor Sie ein neues Flywheel erstellen.

## Erstellen eines Flywheels für ein vorhandenes Modell

Verwenden Sie die [-CreateFlywheel](#)Operation, um ein Flywheel für ein vorhandenes Modell zu erstellen.

### Example

```
aws comprehend create-flywheel \
  --flywheel-name "myFlywheel12" \
  --active-model-arn "modelArn" \
  --data-access-role-arn arn:aws::iam::111122223333:role/testFlywheelDataAccess \
  --data-lake-s3-uri": "https://s3-bucket-endpoint" \
```

Wenn der Vorgang erfolgreich ist, enthält die Antwort den Flywheel-ARN.

```
{
  "FlywheelArn": "arn:aws::comprehend:aws-region:111122223333:flywheel/name",
  "ActiveModelArn": "modelArn"
}
```

## Erstellen eines Flywheels für ein neues Modell

Verwenden Sie die [-CreateFlywheel](#)Operation, um ein Flywheel für ein neues benutzerdefiniertes Klassifizierungsmodell zu erstellen.

### Example

```
aws comprehend create-flywheel \
  --flywheel-name "myFlywheel12" \
  --data-access-role-arn arn:aws::iam::111122223333:role/testFlywheelDataAccess \
  --model-type "DOCUMENT_CLASSIFIER" \
  --data-lake-s3-uri "s3Uri" \
  --task-config file://taskConfig.json
```

Die Datei taskConfig.json enthält den folgenden Inhalt.



```
{
  "LanguageCode": "en",
  "DocumentClassificationConfig": {
    "Mode": "MULTI_LABEL",
    "Labels": ["optimism", "anger"]
  }
}
```

Der API-Antworttext enthält den folgenden Inhalt.

```
{
  "FlywheelArn": "arn:aws::comprehend:aws-region:111122223333:flywheel/name",
  "ActiveModelArn": "modelArn"
}
```

## Beschreiben eines Flywheel

Verwenden Sie die Amazon Comprehend-[DescribeFlywheel](#) Operation, um konfigurierte Informationen zu einem Flywheel abzurufen.

```
aws comprehend describe-flywheel \
  --flywheel-arn "flywheelArn"
```

Der API-Antworttext enthält den folgenden Inhalt.

```
{
  "FlywheelProperties": {
    "FlywheelArn": "arn:aws::comprehend:aws-region:111122223333:flywheel/
myTestFlywheel",
    "DataAccessRoleArn": "arn:aws::iam::111122223333:role/Admin",
    "TaskConfig": {
      "LanguageCode": "en",
      "DocumentClassificationConfig": {
        "Mode": "MULTI_LABEL"
      }
    },
    "DataLakeS3Uri": "s3://my-test-datalake/flywheelbasictest/myTestFlywheel/
schemaVersion=1/20220801T014326Z",
    "Status": "ACTIVE",
    "ModelType": "DOCUMENT_CLASSIFIER",
    "CreationTime": 1659318206.102,
  }
}
```

```
    "LastModifiedTime": 1659318249.05
  }
}
```

## Aktualisieren eines Flywheel

Verwenden Sie die [UpdateFlywheel](#) Operation, um die veränderbaren Konfigurationswerte des Flywheels zu aktualisieren.

Einige Konfigurationsfelder sind JSON-Strukturen mit Unterfeldern. Um ein oder mehrere Unterfelder zu aktualisieren, geben Sie Werte für alle Unterfelder an (Amazon Comprehend setzt den Wert für jedes Unterfeld, das in der Anforderung fehlt, auf Null).

Wenn Sie einen Parameter der obersten Ebene in der `UpdateFlywheel` Anforderung weglassen, ändert Amazon Comprehend die Werte des Parameters oder seiner Unterfelder im Flywheel nicht.

Verwenden Sie die [UntagResource](#) Operationen [TagResource](#) und `untagResource`, um Tags auf dem Flywheel hinzuzufügen oder zu entfernen.

Sie können eine Modellversion hochstufen, indem Sie den `ActiveModelArn` Parameter festlegen, wie im folgenden Beispiel gezeigt.

```
aws comprehend update-flywheel \
  --region aws-region \
  --flywheel-arn "flywheelArn" \
  --active-model-arn "modelArn" \
```

Der API-Antworttext enthält den folgenden Inhalt.

```
{
  "FlywheelArn": "arn:aws::comprehend:aws-region:111122223333:flywheel/name",
  "ActiveModelArn": "modelArn"
}
```

## Löschen eines Flywheel

Verwenden Sie die Amazon Comprehend [DeleteFlywheel](#) Operation, um Flywheels zu löschen.

```
aws comprehend delete-flywheel \
```

```
--flywheel-arn "flywheelArn"
```

Eine erfolgreiche API-Antwort enthält einen leeren Antwortnachrichtentext

## Auflisten der Flywheels

Verwenden Sie die Amazon Comprehend-[ListFlywheels](#) Operation, um eine Liste der Flywheels in der aktuellen Region abzurufen.

```
aws comprehend list-flywheel \  
  --region aws-region \  
  --endpoint-url "uri"
```

Der API-Antworttext enthält den folgenden Inhalt.

```
{  
  "FlywheelSummaryList": [  
    {  
      "FlywheelArn": "arn:aws::comprehend:aws-region:111122223333:flywheel/  
myTestFlywheel",  
      "DataLakeS3Uri": "s3://my-test-datalake/flywheelbasictest/myTestFlywheel/  
schemaVersion=1/20220801T014326Z",  
      "Status": "ACTIVE",  
      "ModelType": "DOCUMENT_CLASSIFIER",  
      "CreationTime": 1659318206.102,  
      "LastModifiedTime": 1659318249.05  
    }  
  ]  
}
```

## Konfigurieren von Datensätzen

Um einem Flywheel beschriftete Trainings- oder Testdaten hinzuzufügen, verwenden Sie die Amazon Comprehend-Konsole oder API, um einen Datensatz zu erstellen.

Sie konfigurieren jeden Datensatz als Trainings- oder Testdaten. Sie verknüpfen den Datensatz mit einem bestimmten Flywheel und einem benutzerdefinierten Modell. Wenn Sie einen Datensatz erstellen, lädt Amazon Comprehend die Daten in den Data Lake des Flywheel hoch. Weitere

Informationen zu Dateiformaten für die Trainingsdaten finden Sie unter [Vorbereitung der Trainingsdaten für Klassifikatoren](#) oder [Vorbereiten von Trainingsdaten für Entity Recognizer](#).

Wenn Sie das Flywheel löschen, löscht Amazon Comprehend die Datensätze. Die hochgeladenen Daten bleiben im Data Lake verfügbar.

## Erstellen eines Datensatzes (Konsole)

### Erstellen eines Datensatzes

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die [Amazon Comprehend-Konsole](#).
2. Wählen Sie im linken Menü Flywheels und dann das Flywheel aus, zu dem Sie die Daten hinzufügen möchten.
3. Wählen Sie die Registerkarte Datensätze aus.
4. Wählen Sie in der Tabelle Trainingsdatensätze oder Testdatensätze die Option Datensatz erstellen aus.
5. Geben Sie unter Datensatzdetails einen Namen für den Datensatz und eine optionale Beschreibung ein.
6. Wählen Sie unter Datenspezifikationen das Datenformat und die Konfigurationsfelder Datensatztyp aus.
7. (Optional) Wählen Sie unter Eingabeformat das Format der Eingabedokumente aus.
8. Geben Sie unter Annotationsspeicherort in S3 den Amazon S3-Speicherort der Annotationsdatei ein.
9. Geben Sie unter Speicherort der Trainingsdaten in S3 den Amazon S3-Speicherort der Dokumentdateien ein.
10. Wählen Sie Erstellen.

## Erstellen eines Datensatzes (API)

Sie können die [CreateDataset](#) Operation verwenden, um einen Datensatz zu erstellen.

### Example

```
aws comprehend create-dataset \  
  --flywheel-arn "myFlywheel12" \  
  --input-format "Text" \  
  --dataset-type "Training" \  
  --s3-train-path "s3://my-bucket/train" \  
  --s3-annotation-path "s3://my-bucket/annotation" \  
  --s3-document-path "s3://my-bucket/document" \  
  --description "my dataset" \  
  --tags "tag1=tag1" "tag2=tag2" \  
  --cli-input-json '{"FlywheelArn": "myFlywheel12", "InputFormat": "Text", "DatasetType": "Training", "S3TrainPath": "s3://my-bucket/train", "S3AnnotationPath": "s3://my-bucket/annotation", "S3DocumentPath": "s3://my-bucket/document", "Description": "my dataset", "Tags": [{"key": "tag1", "value": "tag1"}, {"key": "tag2", "value": "tag2"}]}'
```

```
--dataset-name "my-training-dataset"
--dataset-type "TRAIN"
--description "my training dataset"
--cli-input-json file://inputConfig.json
}
```

Die Datei `inputConfig.json` enthält den folgenden Inhalt.

```
{
  "DataFormat": "COMPREHEND_CSV",
  "DocumentClassifierInputDataConfig": {
    "S3Uri": "s3://my-comprehend-datasets/multilabel_train.csv"
  }
}
```

Um Tags zum Datensatz hinzuzufügen oder zu entfernen, verwenden Sie die [UntagResource](#) Operationen [TagResource](#) und [UntagResource](#).

## Beschreiben eines Datensatzes

Verwenden Sie die Amazon Comprehend-[DescribeDataset](#) Operation, um konfigurierte Informationen zu einem Flywheel abzurufen.

```
aws comprehend describe-dataset \
  --dataset-arn "datasetARN"
```

Die Antwort enthält den folgenden Inhalt.

```
{
  "DatasetProperties": {
    "DatasetArn": "arn:aws::comprehend:aws-region:111122223333:flywheel/
myTestFlywheel/dataset/train-dataset",
    "DatasetName": "train-dataset",
    "DatasetType": "TRAIN",
    "DatasetS3Uri": "s3://my-test-datalake/flywheelbasictest/myTestFlywheel/
schemaVersion=1/20220801T014326Z/datasets/train-dataset/20220801T194844Z",
    "Description": "Good Dataset",
    "Status": "COMPLETED",
    "NumberOfDocuments": 90,
    "CreationTime": 1659383324.297
  }
}
```

}

## Flywheel-Iterationen

Verwenden Sie Flywheel-Iterationen, um neue Modellversionen zu erstellen und zu verwalten.

Themen

- [Iterations-Workflow](#)
- [Verwalten von Iterationen \(Konsole\)](#)
- [Verwalten von Iterationen \(API\)](#)

## Iterations-Workflow

Ein Flywheel beginnt mit einer trainierten Modellversion oder verwendet einen anfänglichen Datensatz, um eine Modellversion zu trainieren.

Wenn Sie neue beschriftete Daten erhalten, trainieren Sie im Laufe der Zeit neue Modellversionen, um die Leistung Ihres Flywheel-Modells zu verbessern. Wenn Sie das Flywheel ausführen, wird eine neue Iteration erstellt, die eine neue Modellversion trainiert und auswertet. Sie können die neue Modellversion hochstufen, wenn ihre Leistung der vorhandenen aktiven Modellversion überlegen ist.

Der Workflow zur Flywheel-Iteration umfasst die folgenden Schritte:

1. Sie erstellen Datensätze für die neuen beschrifteten Daten.
2. Sie führen das Flywheel aus, um eine neue Iteration zu erstellen. Die Iteration folgt diesen Schritten, um eine neue Modellversion zu trainieren und auszuwerten:
  - a. Evaluiert die aktive Modellversion anhand der neuen Daten.
  - b. Trainiert eine neue Modellversion anhand der neuen Daten.
  - c. Speichert die Bewertungs- und Trainingsergebnisse im Data Lake.
  - d. Gibt die F1-Werte für beide Modelle zurück.
3. Nach Abschluss der Iteration können Sie die F1-Werte für das vorhandene aktive Modell und das neue Modell vergleichen.
4. Wenn die neue Modellversion eine hervorragende Leistung aufweist, stufen Sie sie zur aktiven Modellversion hoch. Sie können die [Konsole](#) oder die [API](#) verwenden, um die neue Modellversion hochzustufen.

## Verwalten von Iterationen (Konsole)

Sie können die Konsole verwenden, um eine neue Iteration zu starten und den Status einer laufenden Iteration abzufragen. Sie können auch die Ergebnisse abgeschlossener Iterationen anzeigen.

### Starten einer Flywheel-Iteration (Konsole)

Bevor Sie eine neue Iteration starten können, erstellen Sie einen oder mehrere neue Trainings- oder Testdatensätze. Siehe [Konfigurieren von Datensätzen](#)

### Starten einer Flywheel-Iteration (Konsole)

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die [Amazon Comprehend-Konsole](#).
2. Wählen Sie im linken Menü Flywheels aus.
3. Wählen Sie in der Tabelle Flywheels ein Flywheel aus.
4. Wählen Sie Flywheel ausführen aus.

### Analysieren von Iterationsergebnissen (Konsole)

Nachdem die Flywheel-Iteration ausgeführt wurde, zeigt die Konsole die Ergebnisse in der Tabelle Flywheels-Iterationen an.

### Hochstufen einer neuen Modellversion (Konsole)

Auf der Seite mit den Modelldetails in der Konsole können Sie eine neue Modellversion zur aktiven Modellversion hochstufen.

### Hochstufen einer Flywheel-Modellversion zur aktiven Modellversion (Konsole)

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die [Amazon Comprehend-Konsole](#).
2. Wählen Sie im linken Menü Flywheels aus.
3. Wählen Sie in der Tabelle Flywheels ein Flywheel aus.
4. Wählen Sie in der Tabelle mit den Flywheel-Details die Version aus der Tabelle Flywheels Iterations aus, die hochgestuft werden soll.
5. Wählen Sie Aktives Modell erstellen aus.

## Verwalten von Iterationen (API)

Sie können die Amazon Comprehend API verwenden, um eine neue Iteration zu starten und den Status einer laufenden Iteration abzufragen. Sie können auch die Ergebnisse abgeschlossener Iterationen anzeigen.

### Starten der Flywheel-Iteration (API)

Verwenden Sie die Amazon Comprehend-[StartFlywheelIteration](#) Operation, um eine Flywheel-Iteration zu starten.

```
aws comprehend start-flywheel-iteration \  
  --flywheel-arn "flywheelArn"
```

Die Antwort enthält den folgenden Inhalt.

```
{  
  "FlywheelIterationArn": "arn:aws::comprehend:aws-region:111122223333:flywheel/name"  
}
```

### Hochstufen einer neuen Modellversion (API)

Verwenden Sie die [UpdateFlywheel](#) Operation, um eine Modellversion zur aktiven Modellversion hochzustufen.

Senden Sie die UpdateFlywheel Anforderung mit dem ActiveModelArn Parameter , der auf den ARN der neuen aktiven Modellversion festgelegt ist.

```
aws comprehend update-flywheel \  
  --active-model-arn "modelArn" \  
  --flywheel-arn "flywheelArn"
```

Die Antwort enthält den folgenden Inhalt.

```
{  
  "FlywheelArn": "arn:aws::comprehend:aws-region:111122223333:flywheel/name",  
  "ActiveModelArn": "modelArn"  
}
```



## Beschreiben der Flywheel-Iterationsergebnisse (API)

Die Amazon Comprehend-[DescribeFlywheelIteration](#) Operation gibt Informationen zu einer Iteration zurück, nachdem sie abgeschlossen ist.

```
aws comprehend describe-flywheel-iteration \  
--flywheel-arn "flywheelArn" \  
--flywheel-iteration-id "flywheelIterationId" \  
--region aws-region
```

Die Antwort enthält den folgenden Inhalt.

```
{  
  "FlywheelIterationProperties": {  
    "FlywheelArn": "flywheelArn",  
    "FlywheelIterationId": "iterationId",  
    "CreationTime": <createdAt>,  
    "EndTime": <endedAt>,  
    "Status": <status>,  
    "Message": <message>,  
    "EvaluatedModelArn": "modelArn",  
    "EvaluatedModelMetrics": {  
      "AverageF1Score": <value>,  
      "AveragePrecision": <value>,  
      "AverageRecall": <value>,  
      "AverageAccuracy": <value>  
    },  
    "TrainedModelArn": "modelArn",  
    "TrainedModelMetrics": {  
      "AverageF1Score": <value>,  
      "AveragePrecision": <value>,  
      "AverageRecall": <value>,  
      "AverageAccuracy": <value>  
    }  
  }  
}
```

## Abrufen des Iterationsverlaufs (API)

Verwenden Sie die [ListFlywheelIterationHistory](#) Operation, um Informationen zum Iterationsverlauf abzurufen.

```
aws comprehend list-flywheel-iteration-history \  
--flywheel-arn "flywheelArn"
```

Die Antwort enthält den folgenden Inhalt.

```
{  
  "FlywheelIterationPropertiesList": [  
    {  
      "FlywheelArn": "<flywheelArn>",  
      "FlywheelIterationId": "20220907T214613Z",  
      "CreationTime": 1662587173.224,  
      "EndTime": 1662592043.02,  
      "Status": "<status>",  
      "Message": "<message>",  
      "EvaluatedModelArn": "modelArn",  
      "EvaluatedModelMetrics": {  
        "AverageF1Score": 0.8333333333333333,  
        "AveragePrecision": 0.75,  
        "AverageRecall": 0.9375,  
        "AverageAccuracy": 0.8125  
      },  
      "TrainedModelArn": "modelArn",  
      "TrainedModelMetrics": {  
        "AverageF1Score": 0.865497076023392,  
        "AveragePrecision": 0.7636363636363637,  
        "AverageRecall": 1.0,  
        "AverageAccuracy": 0.84375  
      }  
    }  
  ]  
}
```

## Verwenden von Flywheels für Analysen

Sie können die aktive Modellversion des Flywheels verwenden, um Analysen auf benutzerdefinierte Klassifizierung oder Entitätserkennung durchzuführen. Die aktive Modellversion ist konfigurierbar. Sie können die [???Konsole](#) oder die [UpdateFlywheel](#) API-Operation verwenden, um eine neue Version des Modells als aktive Modellversion festzulegen.

Um das Flywheel zu verwenden, geben Sie bei der Konfiguration der Analyseaufgabe den Flywheel-ARN anstelle eines benutzerdefinierten Modell-ARN an. Amazon Comprehend führt die Analyse mit der aktiven Modellversion des Flywheels aus.

## Echtzeitanalyse

Sie verwenden einen Endpunkt, um Echtzeitanalysen durchzuführen. Wenn Sie einen Endpunkt erstellen oder aktualisieren, können Sie ihn mit dem Flywheel-ARN anstelle eines Modell-ARN konfigurieren. Wenn Sie die Echtzeitanalyse ausführen, wählen Sie den Endpunkt aus, der dem Flywheel zugeordnet ist. Amazon Comprehend führt die Analyse mit der aktiven Modellversion des Flywheels durch.

Wenn Sie verwenden [UpdateFlywheel](#), um eine neue aktive Modellversion für das Flywheel festzulegen, wird der Endpunkt automatisch aktualisiert, um die neue aktive Modellversion zu verwenden. Wenn Sie nicht möchten, dass der Endpunkt automatisch aktualisiert wird, konfigurieren Sie den Endpunkt (mit [UpdateEndpoint](#)), um den ARN der Modellversion direkt zu verwenden. Der Endpunkt verwendet diese Modellversion weiterhin, wenn sich die aktive Modellversion des Flywheel ändert.

Verwenden Sie für die benutzerdefinierte Klassifizierung die [ClassifyDocument](#)-API-Operation. Verwenden Sie für die Erkennung benutzerdefinierter Entitäten die [DetectEntities](#)-API-Anfrage. Geben Sie den Endpunkt des Flywheels im EndpointArn Parameter an.

Sie können die Konsole auch verwenden, um Echtzeitanalysen für [benutzerdefinierte Klassifizierung](#) oder [benutzerdefinierte Entitätserkennung](#) durchzuführen.

## Asynchrone Aufträge

Verwenden Sie für die benutzerdefinierte Klassifizierung die [StartDocumentClassificationJob](#) -API-Anfrage, um einen asynchronen Auftrag zu starten. Geben Sie den FlywheelArn Parameter anstelle von anDocumentClassifierArn.

Verwenden Sie für die Erkennung benutzerdefinierter Entitäten die [StartEntitiesDetectionJob](#)-API-Anfrage. Geben Sie den FlywheelArn Parameter anstelle von anEntityRecognizerArn.

Sie können die Konsole verwenden, um asynchrone Analyseaufträge für die [benutzerdefinierte Klassifizierung](#) oder [die Erkennung benutzerdefinierter Entitäten](#) auszuführen. Wenn Sie den Auftrag erstellen, geben Sie den Flywheel-ARN in das Feld Recognizer-Modell oder Classifier-Modell ein.

# Verwalten von Amazon Comprehend-Endpunkten

In Amazon Comprehend stellen Endpunkte Ihre benutzerdefinierten Modelle für die Echtzeitklassifizierung oder Entitätserkennung zur Verfügung. Nachdem Sie einen Endpunkt erstellt haben, können Sie Änderungen daran vornehmen, wenn sich Ihre Geschäftsanforderungen weiterentwickeln. Sie können beispielsweise Ihre Endpunktauslastung überwachen und Auto Scaling anwenden, um die Endpunktbereitstellung automatisch an Ihre Kapazitätsanforderungen anzupassen. Sie können alle Ihre Endpunkte in einer einzigen Ansicht verwalten. Wenn Sie einen Endpunkt nicht mehr benötigen, können Sie ihn löschen, um Kosten zu sparen.

Bevor Sie einen Endpunkt verwalten können, müssen Sie einen erstellen. Weitere Informationen finden Sie in den folgenden Verfahren:

- [Einen Endpunkt für die benutzerdefinierte Klassifizierung erstellen](#)
- [Erstellen eines Endpunkts für die Erkennung benutzerdefinierter Entitäten](#)

## Themen

- [Übersicht über Amazon Comprehend-Endpunkte](#)
- [Verwenden von Amazon Comprehend-Endpunkten](#)
- [Überwachen von Amazon Comprehend-Endpunkten](#)
- [Aktualisieren von Amazon Comprehend-Endpunkten](#)
- [Verwenden von Trusted Advisor mit Amazon Comprehend](#)
- [Löschen von Amazon Comprehend-Endpunkten](#)
- [Auto Scaling mit Endpunkten](#)

## Übersicht über Amazon Comprehend-Endpunkte

Auf der Seite Endpunkte der Amazon Comprehend-Konsole erhalten Sie eine globale Ansicht Ihrer Endpunkte. Auf der Übersichtsseite Endpunkte können Sie alle Endpunkte an einem Ort anzeigen, um Ihre Endpunktnutzung im Vergleich zu Ihrer tatsächlichen Ressourcennutzung zu verstehen. Oben rechts auf der Endpunktseite können Sie angeben, welche Endpunkte Sie anzeigen möchten – alle, benutzerdefinierte Classifier-Endpunkte oder Ihre benutzerdefinierten Entitätsendpunkte.

Auf dieser Seite können Sie Endpunkte erstellen, aktualisieren, überwachen und löschen. Im Abschnitt Endpunktübersicht können Sie eine Liste Ihrer Endpunkte, die benutzerdefinierten Modelle,

die die Endpunkte hosten, ihre Erstellungszeit, den bereitgestellten Durchsatz und den Status des Endpunkts anzeigen. Wenn Sie einen bestimmten Endpunkt aus der Endpunktübersichtstabelle auswählen, werden die Endpunktdetails angezeigt.

Wenn Sie ein [AWS Business Support](#)- oder [AWS Enterprise Support](#)-Kunde sind, haben Sie auch Zugriff auf Trusted Advisor-Prüfungen, die für Ihre Endpunkte spezifisch sind. Weitere Informationen hierzu finden Sie unter [Verwenden von Trusted Advisor mit Amazon Comprehend](#). Eine vollständige Liste der Prüfungen und Beschreibungen finden Sie unter [Trusted Advisor Best Practices](#).

Weitere Informationen zur Verwaltung Ihrer Endpunkte finden Sie in den folgenden Themen.

- [Verwenden von Amazon Comprehend-Endpunkten](#)
- [Überwachen von Amazon Comprehend-Endpunkten](#)
- [Aktualisieren von Amazon Comprehend-Endpunkten](#)
- [Verwenden von Trusted Advisor mit Amazon Comprehend](#)
- [Löschen von Amazon Comprehend-Endpunkten](#)

#### Important

Die Kosten für die benutzerdefinierte Echtzeitklassifizierung basieren sowohl auf dem von Ihnen festgelegten Durchsatz als auch auf der Zeitdauer, für die der Endpunkt aktiv ist. Wenn Sie den Endpunkt nicht mehr oder über einen längeren Zeitraum nicht verwenden, sollten Sie eine Auto-Scaling-Richtlinie einrichten, um Ihre Kosten zu senken. Oder wenn Sie keinen Endpunkt mehr verwenden, können Sie den Endpunkt löschen, um zusätzliche Kosten zu vermeiden. Weitere Informationen finden Sie unter [Auto Scaling mit Endpunkten](#).

## Verwenden von Amazon Comprehend-Endpunkten

Sie erstellen einen Endpunkt, um Echtzeitanalysen mit einem benutzerdefinierten Modell durchzuführen. Ein Endpunkt umfasst verwaltete Ressourcen, die Ihr benutzerdefiniertes Modell für Echtzeit-Inferenzen verfügbar machen.

Amazon Comprehend weist einem Endpunkt mithilfe von Inferenzeinheiten (IU) den Durchsatz zu. Ein Bol steht für einen Datendurchsatz von 100 Zeichen pro Sekunde. Sie können dem Endpunkt bis zu 10 Inferenzeinheiten bereitstellen. Sie können den Endpunktdurchsatz entweder nach oben oder unten skalieren, indem Sie den Endpunkt aktualisieren.

Wenn Ihre Eingabedokumente halbstrukturierte Dokumente oder Bilddateien enthalten, bezieht sich der Durchsatz von 100 Zeichen pro Sekunde auf die aus der Eingabedatei extrahierten Zeichen. Die Anzahl der IUs, die Sie für einen Endpunkt bereitstellen, hängt von der Zeichendichte der Eingabedokumente ab.

Die - [ClassifyDocument](#) und [DetectEntities](#)-API-Antworten enthalten die Zeichenanzahl für jede Eingabeseite. Sie können diese Informationen verwenden, um die Anzahl der Inferenzeinheiten zu schätzen, die bereitgestellt werden sollen, um den gewünschten Durchsatz zu erreichen.

Nachdem Sie Ihre Echtzeitanalyse abgeschlossen haben, löschen Sie den Endpunkt, da die Gebühr dafür so lange andauert, wie er aktiv ist. Sie können einen weiteren Endpunkt erstellen, wenn Sie bereit sind, weitere Echtzeitanalysen durchzuführen.

Weitere Informationen zu Endpunktkosten finden Sie unter [Amazon Comprehend – Preise](#).

Nachdem Sie einen Endpunkt erstellt haben, können Sie ihn mit Amazon überwachen CloudWatch, ihn aktualisieren, um seine Inferenzeinheiten zu ändern, oder ihn löschen, wenn er nicht mehr benötigt wird. Weitere Informationen finden Sie unter [Überwachen von Amazon Comprehend-Endpunkten](#).

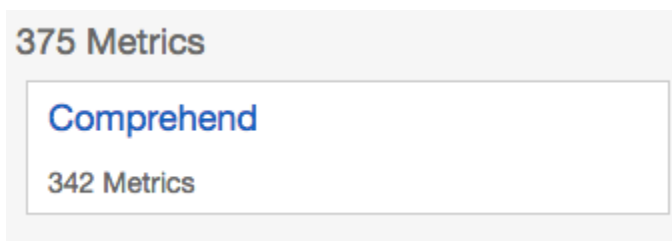
## Überwachen von Amazon Comprehend-Endpunkten

Sie können den Durchsatz Ihres Endpunkts anpassen, indem Sie die Anzahl der Inferenzeinheiten (IUs) erhöhen oder verringern. Weitere Informationen zum Aktualisieren Ihres Endpunkts finden Sie unter [the section called “Aktualisieren von Endpunkten”](#).

Sie können bestimmen, wie Sie den Durchsatz Ihres Endpunkts am besten anpassen können, indem Sie dessen Nutzung mit der Amazon- CloudWatch Konsole überwachen.

Überwachen Ihrer Endpunktnutzung mit CloudWatch

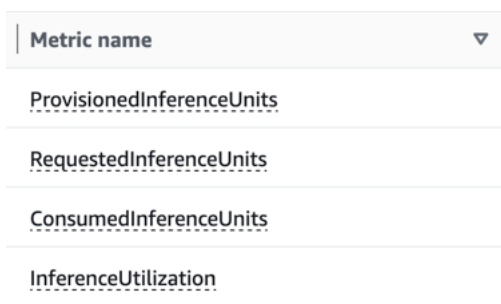
1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die [CloudWatch -Konsole](#).
2. Wählen Sie links Metriken und dann Alle Metriken aus.
3. Wählen Sie unter Alle Metriken die Option Comprehend aus.



- Die CloudWatch Konsole zeigt die Dimensionen für die Comprehend-Metriken an. Wählen Sie die EndpointArn Dimension aus.



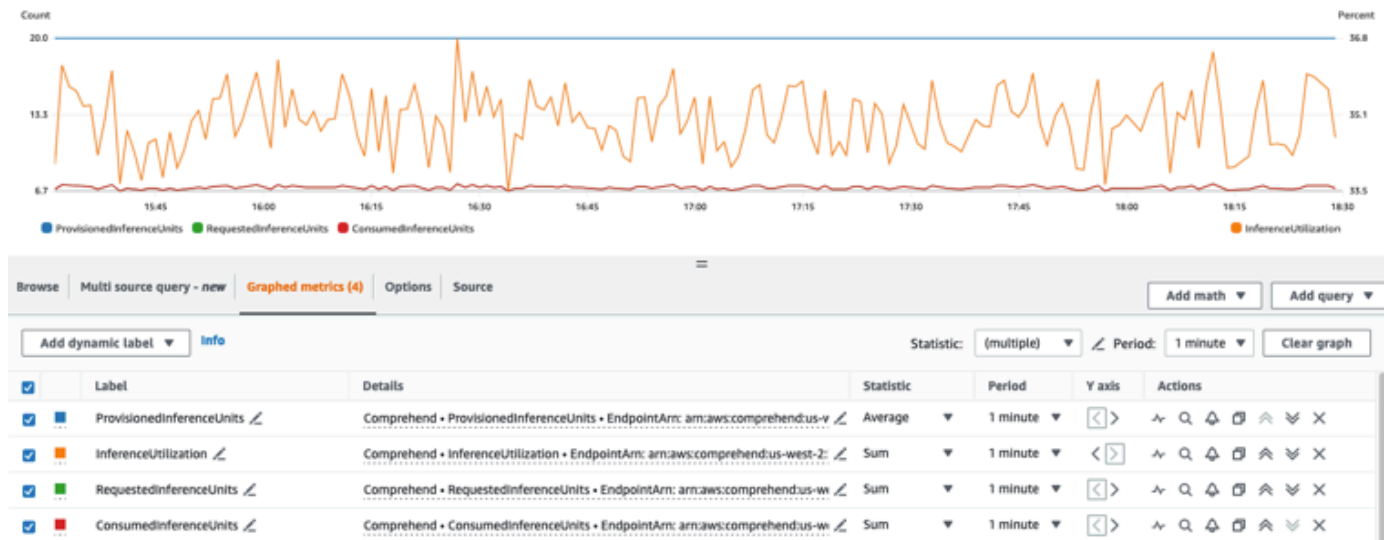
Die Konsole zeigt ProvisionedInferenceUnits, RequestedInferenceUnitsConsumedInferenceUnits, und InferenceUtilization für jeden Ihrer Endpunkte an.



Wählen Sie die vier Metriken aus und navigieren Sie zur Registerkarte Grafische Metriken.

- Legen Sie die Statistikspalten für RequestedInferenceUnits und ConsumedInferenceUnits auf Summe fest.
- Legen Sie die Spalte Statistik für InferenceUtilization auf Summe fest.
- Legen Sie die Spalte Statistik für ProvisionedInferenceUnits auf Durchschnitt fest.
- Ändern Sie die Spalte Zeitraum für alle Metriken auf 1 Minute.
- Wählen Sie InferenceUtilization und dann den Pfeil aus, um ihn in eine separate Y-Achse zu verschieben.

Ihr Diagramm ist für die Analyse bereit.



Basierend auf den CloudWatch Metriken können Sie auch Auto Scaling einrichten, um den Durchsatz Ihres Endpunkts automatisch anzupassen. Weitere Informationen zur Verwendung von Auto Scaling mit Ihren Endpunkten finden Sie unter [Auto Scaling mit Endpunkten](#).

- **ProvisionedInferenceUnits** – Diese Metrik stellt die Anzahl der durchschnittlichen bereitgestellten IUs zum Zeitpunkt der Anforderung dar.
- **RequestedInferenceUnits** – Dies basiert auf der Nutzung jeder Anforderung, die an den Service gesendet wurde, der zur Verarbeitung gesendet wurde. Dies kann hilfreich sein, um die zur Verarbeitung gesendete Anfrage mit der tatsächlich verarbeiteten Anfrage zu vergleichen, ohne dass eine Drosselung () auftrittConsumedInferenceUnits. Der Wert für diese Metrik wird berechnet, indem die Anzahl der Zeichen, die zur Verarbeitung gesendet werden, durch die Anzahl der Zeichen dividiert wird, die in einer Minute für 1 verarbeitet werden können.
- **ConsumedInferenceUnits** – Dies basiert auf der Verwendung jeder an den Service übermittelten Anfrage, die erfolgreich verarbeitet (nicht gedrosselt) wurde. Dies kann hilfreich sein, wenn Sie vergleichen, was Sie verbrauchen, mit Ihren bereitgestelltenIUs. Der Wert für diese Metrik wird berechnet, indem die Anzahl der verarbeiteten Zeichen durch die Anzahl der Zeichen dividiert wird, die in einer Minute für 1 verarbeitet werden können.
- **InferenceUtilization** – Dies wird pro Anforderung ausgegeben. Dieser Wert wird berechnet, indem die in definierten IUs durch ConsumedInferenceUnits dividiert ProvisionedInferenceUnits und in einen Prozentsatz von 100 konvertiert werden.



**Note**

Alle Metriken werden nur für erfolgreiche Anforderungen ausgegeben. Die Metrik wird nicht angezeigt, wenn sie aus einer Anforderung stammt, die gedrosselt wurde oder mit einem internen Serverfehler oder einem Kundenfehler fehlschlägt.

## Aktualisieren von Amazon Comprehend-Endpunkten


Häufig ändert sich der Durchsatz, den Sie nach dem Erstellen eines Endpunkts benötigen, oder Ihre erste Schätzung Ihrer Anforderungen ändert sich. In diesem Fall ist es möglicherweise erforderlich, Ihren Endpunkt zu aktualisieren, um den Durchsatz nach oben oder unten anzupassen. Der Durchsatz hängt von der Anzahl der Inferenzeinheiten ab, mit denen Sie Ihren Endpunkt bereitgestellt haben. Jede Inferenzeinheit stellt einen Durchsatz von 100 Zeichen pro Sekunde für bis zu 2 Dokumente pro Sekunde dar. Möglicherweise möchten Sie auch die Version des Modells aktualisieren, die dem Endpunkt zugeordnet ist. Wenn Sie einen Endpunkt bearbeiten, können Sie eine andere Version des Modells für den Endpunkt auswählen.

Es kann auch hilfreich sein, Ihrem Endpunkt Tags hinzuzufügen, um sie organisiert zu halten. Dies kann auch während der Aktualisierung Ihres Endpunkts erfolgen. Weitere Informationen zu Endpunkten finden Sie unter [Markieren Ihrer -Ressourcen](#)

So aktualisieren Sie einen Endpunkt (Konsole)

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon Comprehend-Konsole unter <https://console.aws.amazon.com/comprehend/>
2. Wählen Sie im linken Menü Endpunkte aus.
3. Wählen Sie in der Liste Classifiers den Namen des benutzerdefinierten Modells aus, von dem Sie den Endpunkt aktualisieren möchten, und folgen Sie dem Link. Die Seite mit den Modelldetails wird angezeigt.
4. Wählen Sie auf der Seite mit den Modelldetails die Versionsdetails aus. Die Liste der Endpunkte wird angezeigt.
5. Aktivieren Sie das Kontrollkästchen Endpunkt für Ihren Endpunkt. Wählen Sie oben rechts in der Tabelle Endpunkte das Symbol Aktionen aus.
6. Wählen Sie Bearbeiten aus. Sie können bereitgestellte IUs aktualisieren und Tags bearbeiten.
7. Speichern Sie Ihre Änderungen.

8. Um die Anzahl der Inferenzeinheiten zu bearbeiten, mit denen der Endpunkt bereitgestellt wird, wählen Sie Bearbeiten aus.
9. Geben Sie die aktualisierte Anzahl von Inferenzeinheiten ein, die dem Endpunkt zugewiesen werden sollen. Jede Einheit stellt einen Durchsatz von 100 Zeichen pro Sekunde dar. Sie können maximal 10 Inferenzeinheiten pro Endpunkt zuweisen.

 Note

Die Kosten für die Verwendung eines Endpunkts basieren auf der Betriebszeit und dem Durchsatz (basierend auf der Anzahl der Inferenzeinheiten). Eine Erhöhung der Anzahl der Inferenzeinheiten erhöht somit die Betriebskosten. Weitere Informationen finden Sie unter [Amazon Comprehend – Preise](#).

10. Wählen Sie Endpunkt bearbeiten aus. Die Seite mit den Endpunktdetails wird angezeigt.
11. Bestätigen Sie, dass der Endpunkt aktualisiert wird, indem Sie den Modellnamen aus den Breadcrumbs oben auf der Seite auswählen. Navigieren Sie auf der Detailseite des benutzerdefinierten Modells zur Liste Endpunkte und überprüfen Sie, ob neben dem Endpunkt die Option Aktualisieren angezeigt wird. Wenn das Update abgeschlossen ist, wird Bereit angezeigt.

Das folgende Beispiel zeigt die Verwendung der `-UpdateEndpointOperation` mit der AWS-CLI.

Das Beispiel ist für Unix, Linux und macOS formatiert. Ersetzen Sie unter Windows den umgekehrten Schrägstrich (`\`), das Unix-Fortsetzungszeichen, am Ende jeder Zeile durch ein Caret-Zeichen oder Zirkumflex (`^`).

```
aws comprehend update-endpoint \  
  --desired-inference-units updated number of inference units \  
  --desired-model-arn arn:aws:comprehend:region:account-id:model type/model name \  
 \  
  --desired-data-access-role-arn arn:aws:iam:account id:role/role name \  
  --endpoint-arn arn:aws:comprehend:region:account id:endpoint/endpoint name
```

Wenn die Aktion erfolgreich ist, antwortet Amazon Comprehend mit einer HTTP 200-Antwort mit einem leeren HTTP-Text.

12. Um das an Ihren Endpunkt angefügte benutzerdefinierte Modell zu bearbeiten, navigieren Sie auf der Seite mit den benutzerdefinierten Modelldetails zur Liste Endpunkte.
13. Wählen Sie den Endpunkt aus, den Sie ändern möchten, und wählen Sie Bearbeiten aus.

14. Auf der Seite mit den Endpunkteinstellungen können Sie unter Classifier-Modell auswählen oder Recognizer-Modell je nach Endpunkt auswählen im Dropdown-Menü nach einem Modell suchen. Wählen Sie das gewünschte Modell aus.
15. Unter Version auswählen können Sie nach der gewünschten Modellversion suchen. Wählen Sie die Version aus.
16. Wählen Sie zum Speichern Endpunkt bearbeiten aus.

## Verwenden von Trusted Advisor mit Amazon Comprehend

AWS Trusted Advisor ist ein Online-Tool, das Empfehlungen zur Verfügung stellt, mit denen Sie Ihre Ressourcen gemäß den AWS bewährten Methoden bereitstellen können.

Wenn Sie über einen Basic- oder Developer-Supportplan verfügen, können Sie die Trusted Advisor Konsole verwenden, um auf alle Prüfungen in der Kategorie Service Limits und auf sechs Prüfungen in der Kategorie Sicherheit zuzugreifen. Wenn Sie über einen Business- oder Enterprise-Supportplan verfügen, können Sie die Trusted Advisor Konsole und die [AWS Support API](#) verwenden, um auf alle Trusted Advisor Prüfungen zuzugreifen.

Amazon Comprehend unterstützt die folgenden Trusted Advisor Prüfungen, um Kunden bei der Optimierung der Kosten und der Sicherheit ihrer Amazon Comprehend-Endpunkte zu unterstützen, indem umsetzbare Empfehlungen bereitgestellt werden.

## Nicht ausgelastete Endpunkte von Amazon Comprehend

Die Prüfung nicht ausgelasteter Endpunkte von Amazon Comprehend bewertet die Durchsatzkonfiguration Ihrer Endpunkte. Diese Prüfung warnt Sie, wenn Endpunkte nicht aktiv für Echtzeit-Inferenzanforderungen genutzt werden. Ein Endpunkt, der länger als 15 Tage nicht verwendet wird, gilt als nicht ausgelastet. Für alle Endpunkte fallen Gebühren sowohl auf der Grundlage des festgelegten Durchsatzes als auch auf der Zeitdauer an, für die der Endpunkt aktiv ist. Für den Endpunkt, der in den letzten 15 Tagen nicht verwendet wurde, empfehlen wir Ihnen, mithilfe von [Application Autoscaling](#) eine Skalierungsrichtlinie für die Ressource zu definieren. Für einen Endpunkt, der in den letzten 30 Tagen nicht verwendet wurde und für den eine Auto-Scaling-Richtlinie definiert ist, empfehlen wir, asynchrone Inferenz zu verwenden oder sie zu löschen. Diese Prüfergebnisse werden einmal täglich automatisch aktualisiert und können in der CostOptimization Kategorie auf der Trusted Advisor Konsole angezeigt werden.

So zeigen Sie den Auslastungsstatus all Ihrer Endpunkte und die entsprechenden Empfehlungen an

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die - Trusted Advisor Konsole.
2. Wählen Sie im Navigationsbereich die CostOptimization Prüfungskategorie aus.
3. Auf der Kategorieseite können Sie die Zusammenfassung für jede Prüfungskategorie anzeigen:
  - Empfohlene Aktion (rot) – Trusted Advisor empfiehlt eine Aktion für die Prüfung.
  - Empfohlene Untersuchung (Gelb) – Trusted Advisor stellt ein mögliches Problem bei der Prüfung fest.
  - Keine Probleme erkannt (grün) – erkennt Trusted Advisor kein Problem für die Prüfung.
  - Ausgeschlossene Elemente (grau) – Die Anzahl der Prüfungen, bei denen Elemente ausgeschlossen wurden, z. B. Ressourcen, die eine Prüfung ignorieren soll.
4. Wählen Sie Amazon Comprehend Unutilized Endpoints check aus, um die Beschreibung der Prüfung und die folgenden Details anzuzeigen:
  - Alarmkriterien – Beschreibt den Schwellenwert, ab dem eine Prüfung ihren Status ändert.
  - Empfohlene Aktion – Beschreibt die empfohlenen Maßnahmen für diese Prüfung.
  - Ressourcentabelle: Eine Tabelle, die Ihre Endpunktdetails und den Status für jede auf der Grundlage Ihrer Empfehlungen auflistet.
5. Wenn in der Tabelle Ressource ein Endpunkt aufgrund einer Warnung, die in den letzten 30 Tagen nicht verwendet wurde, mit einer Untersuchungsempfehlung gekennzeichnet ist, können Sie in der Amazon Comprehend-Konsole zur Seite Endpunktdetails navigieren.
  - Wenn Sie diesen Endpunkt nicht mehr verwenden möchten, wählen Sie Löschen aus.
  - Zum Bestätigen des Löschvorgangs wählen Sie erneut Delete (Löschen) aus. Die Seite mit den benutzerdefinierten Modelldetails wird angezeigt. Vergewissern Sie sich, dass neben dem gelöschten Endpunkt das Löschen angezeigt wird. Nach dem Löschen wird der Endpunkt aus der Liste Endpunkte entfernt.
6. Wenn in der Tabelle Resource in der Trusted Advisor Konsole ein Endpunkt mit dem Status Inspect Recommended gekennzeichnet ist, weil er in den letzten 15 Tagen nicht verwendet wurde, und wenn er AutoScaling deaktiviert wurde, können Sie zur Seite Endpoint Details in der Amazon Comprehend-Konsole navigieren, um den Endpunkt anzupassen.
  - Wenn Sie den für diesen Endpunkt konfigurierten Durchsatz reduzieren möchten, klicken Sie auf Bearbeiten. Geben Sie die aktualisierte Anzahl von Inferenzeinheiten ein, die dem Endpunkt zugewiesen werden sollen, aktivieren Sie dann das Kontrollkästchen,

um zu bestätigen, und wählen Sie dann Endpunkt bearbeiten aus. Wenn das Update abgeschlossen ist, wird der Status als Bereit angezeigt.

- Wenn Sie die Endpunktbereitstellung auf Ihrem Endpunkt automatisch festlegen möchten, anstatt die Durchsatzkonfiguration manuell anzupassen, empfehlen wir Ihnen, Application Autoscaling zu verwenden.
7. Wenn Trusted Advisor ein Endpunkt in der Konsole mit dem Status Keine Probleme erkannt aus dem Grund Aktiv verwendet gekennzeichnet wird, bedeutet dies, dass der Endpunkt aktiv für die Ausführung von Echtzeit-Inferenzanfragen genutzt wird und keine Aktionen empfohlen werden.

Hier ist ein Beispiel, das die CostOptimization Kategorieansicht in der Trusted Advisor Konsole zeigt:

**Cost Optimization** Refresh all checks Download all checks

Choose a check name to see recommendations for ways to help save money for your AWS account. Trusted Advisor might recommend that you delete unused and idle resources, or use reserved capacity.

**Cost Optimization Checks**

Potential Monthly Savings: **\$14,881.82**

0 Action recommended | 13 Investigation recommended | 3 No problems detected | 0 Excluded items

Filter by tag [Learn more about using tags](#)

Tag Key:  Tag Value:  Reset Apply filter View by: All checks

- Amazon EC2 Reserved Instances Optimization** Refreshed: a day ago

A significant part of using AWS involves balancing your Reserved Instance (RI) usage and your On-Demand instance usage. Estimated monthly savings with one year RI term: \$329.91 (24.0%). Estimated monthly savings with three year RI term: \$530.07 (38.0%)
- Amazon RDS Idle DB Instances** Refreshed: a day ago

Checks the configuration of your Amazon Relational Database Service (Amazon RDS) for any DB instances that appear to be idle. 28 of 29 DB instances appear to be Idle. Monthly savings of up to \$3,744 are available by minimizing idle DB Instances.
- Amazon Redshift Reserved Node Optimization** Refreshed: a day ago

Checks your usage of Redshift and provides recommendations on purchase of Reserved Nodes to help reduce costs incurred from using Redshift On-Demand. Estimated monthly savings with one year Reserved Node term: \$1,069.77 (32.0%). Estimated monthly savings with three year Reserved Node term: \$2,024.31 (60.0%).

## Zugriffsrisiko für Amazon Comprehend-Endpunkte

Die Amazon Amazon Comprehend-Endpunkt-Zugriffsrisikoprüfung bewertet die AWS Key Management Service (AWS KMS)-Schlüsselberechtigungen für einen Endpunkt, bei dem das zugrunde liegende Modell mit vom Kunden verwalteten Schlüsseln verschlüsselt wurde. Wenn der vom Kunden verwaltete Schlüssel deaktiviert ist oder die Schlüsselrichtlinie geändert wurde, um die zulässigen Berechtigungen für Amazon Comprehend zu ändern, kann die Verfügbarkeit des Endpunkts beeinträchtigt werden. Wenn der Schlüssel deaktiviert wurde, empfehlen

wir, ihn zu aktivieren. Wenn die Schlüsselrichtlinie geändert wurde und Sie diesen Endpunkt weiterhin verwenden möchten, empfehlen wir Ihnen, die Schlüsselrichtlinie zu aktualisieren. Die Prüfungsergebnisse werden mehrmals am Tag automatisch aktualisiert. Diese Prüfung kann unter der Kategorie Fehlertoleranz der - Trusted Advisor Konsole angezeigt werden.

So zeigen Sie den AWS KMS Schlüsselstatus Ihrer Amazon Comprehend-Endpunkte an

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die - Trusted Advisor Konsole.
2. Wählen Sie im Navigationsbereich die FaultTolerance Prüfungskategorie aus.
3. Auf der Kategorieseite können Sie die Zusammenfassung für jede Prüfungskategorie anzeigen:
  - Empfohlene Aktion (rot) – Trusted Advisor empfiehlt eine Aktion für die Prüfung.
  - Untersuchung empfohlen (EoL) – Trusted Advisor erkennt ein mögliches Problem für die Prüfung.
  - Keine Probleme erkannt (grün) – erkennt Trusted Advisor kein Problem bei der Prüfung.
  - Ausgeschlossene Elemente (grau) – Die Anzahl der Prüfungen, bei denen Elemente ausgeschlossen wurden, z. B. Ressourcen, die bei einer Prüfung nicht berücksichtigt werden sollen.
4. Wählen Sie Amazon Comprehend Endpoint Access Risk Check und Sie können die Beschreibung der Prüfung sowie die folgenden Details anzeigen:
  - Warnungskriterien – Beschreibt den Schwellenwert, an dem sich der Status einer Prüfung ändert.
  - Empfohlene Aktion – Beschreibt die empfohlenen Maßnahmen für diese Prüfung.
  - Ressourcentabelle: Eine Tabelle, in der die Details Ihres verschlüsselten KMS-Endpunkts und der Status für jeden Endpunkt aufgeführt sind, basierend auf den empfohlenen Aktionen.
5. Wenn in der Tabelle Ressource ein Endpunkt mit dem Status Aktion empfohlen gekennzeichnet ist, wählen Sie den Link in der KeyId Spalte KMS aus und Sie werden zur entsprechenden AWS KMS Schlüsselseite weitergeleitet.
  - Um einen deaktivierten AWS KMS Schlüssel zu aktivieren, wählen Sie Schlüsselaktionen und dann Aktivieren aus.
  - Wenn der Schlüsselstatus als Aktiviert aufgeführt ist, aktualisieren Sie die Schlüsselrichtlinie, indem Sie im Abschnitt Schlüsselrichtlinie die Option Zur Richtlinienansicht wechseln auswählen. Bearbeiten Sie das Schlüsselrichtliniendokument, um Amazon Comprehend die

erforderlichen Berechtigungen bereitzustellen, und wählen Sie dann Änderungen speichern aus.

Hier ist ein Beispiel für die FaultTolerance Kategorieansicht in der Trusted Advisor Konsole:

**Fault tolerance checks**

⊗ 0 Info    ⚠ 0 Info    ✓ 1 Info    ⊖ 0 Info

Action recommended    Investigation recommended    No problems detected    Excluded items

Filter by tag [Learn more about using tags](#)

Tag Key    Tag Value    Reset    Apply filter

View by: All checks

- ▶ ✓ **AWS Lambda VPC-enabled Functions without Multi-AZ Redundancy** Refreshed: 11 hours ago  
Checks for VPC-enabled Lambda functions that are vulnerable to service interruption in a single availability zone.
- ▶ ⊖ **Amazon Aurora DB Instance Accessibility**  
Checks for cases where an Amazon Aurora DB cluster has both private and public instances.
- ▶ ⊖ **Amazon EBS Snapshots**  
Checks the age of the snapshots for your Amazon Elastic Block Store (Amazon EBS) volumes (available or in-use).
- ▶ ⊖ **Amazon EC2 Availability Zone Balance**  
Checks the distribution of Amazon Elastic Compute Cloud (Amazon EC2) instances across Availability Zones in a region.

Diese Prüfungen und ihre Ergebnisse können auch angezeigt werden, indem Sie auf den - Trusted Advisor Abschnitt der AWS Support -API verweisen.

Weitere Informationen zum Einrichten von Alarmen mit CloudWatch finden Sie unter: [Erstellen von Trusted Advisor Alarmen mit CloudWatch](#). Eine vollständige Reihe von Trusted Advisor Best-Practice-Prüfungen finden Sie unter: [AWS Trusted Advisor Checkliste für bewährte Methoden](#).

## Löschen von Amazon Comprehend-Endpunkten

Sobald Sie Ihren Endpunkt nicht mehr benötigen, sollten Sie ihn löschen, damit Ihnen keine Kosten dafür entstehen. Sie können ganz einfach einen anderen Endpunkt erstellen, wenn Sie ihn benötigen, und zwar im Abschnitt Endpunkte.

Löschen Sie einen Endpunkt wie folgt (Konsole):

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon Comprehend-Konsole unter <https://console.aws.amazon.com/comprehend/>
2. Wählen Sie im linken Menü Endpunkte aus.
3. Suchen Sie in der Tabelle Endpunkte den Endpunkt, den Sie löschen möchten. Sie können alle Endpunkte suchen oder filtern, um den gewünschten zu finden.
4. Aktivieren Sie das Kontrollkästchen Endpunkt für den Endpunkt, den Sie löschen möchten. Wählen Sie oben rechts in der Tabelle Endpunkte das Symbol Aktionen aus.
5. Wählen Sie Löschen aus.
6. Zum Bestätigen des Löschvorgangs wählen Sie erneut Delete (Löschen) aus. Die Seite Endpunkte wird angezeigt. Vergewissern Sie sich, dass neben dem gelöschten Endpunkt Löschen angezeigt wird. Wenn er gelöscht wird, wird der Endpunkt aus der Liste Endpunkte entfernt.

So löschen Sie einen Endpunkt (AWS CLI)

Das folgende Beispiel zeigt die Verwendung der `-DeleteEndpointOperation` mit der AWS-CLI.

Das Beispiel ist für Unix, Linux und macOS formatiert. Ersetzen Sie unter Windows den umgekehrten Schrägstrich (`\`), das Unix-Fortsetzungszeichen, am Ende jeder Zeile durch ein Caret-Zeichen oder Zirkumflex (`^`).

```
aws comprehend delete-endpoint \  
  --endpoint-arn arn:aws:comprehend:region:account-id endpoint/endpoint name
```

Wenn die Aktion erfolgreich ist, antwortet Amazon Comprehend mit einer HTTP 200-Antwort mit einem leeren HTTP-Text.

## Auto Scaling mit Endpunkten

Anstatt die Anzahl der Inferenzeinheiten, die für Ihre Endpunkte zur Dokumentenklassifizierung und Entity Recognizer bereitgestellt werden, manuell anzupassen, können Sie Auto Scaling verwenden, um die Endpunktbereitstellung automatisch an Ihre Kapazitätsanforderungen anzupassen.



Es gibt zwei Möglichkeiten, Auto Scaling zu verwenden, um die Anzahl der für Ihren Endpunkt bereitgestellten Inferenzeinheiten anzupassen:

- [Zielverfolgung](#): Legen Sie Auto Scaling fest, um die Endpunktbereitstellung an Kapazitätsanforderungen anzupassen, je nach Nutzung.
- [Geplante Skalierung](#): Legen Sie Auto Scaling fest, um die Endpunktbereitstellung an die Kapazitätsanforderungen nach einem bestimmten Zeitplan anzupassen.

Sie können Auto Scaling nur mit der AWS Command Line Interface (AWS CLI) festlegen. Weitere Informationen zum Auto Scaling finden Sie unter [Was ist Application Auto Scaling?](#)

## Zielverfolgung

Mit der Zielverfolgung können Sie die Endpunktbereitstellung an Ihre Kapazitätsanforderungen an die Nutzung anpassen. Die Anzahl der Inferenzeinheiten wird automatisch angepasst, sodass die genutzte Kapazität innerhalb eines Zielprozentsatzes der bereitgestellten Kapazität liegt. Sie können die Zielverfolgung verwenden, um vorübergehende Nutzungsspitzen für Ihre Endpunkte zur Dokumentenklassifizierung und Entity Recognizer-Endpunkte zu bewältigen. Weitere Informationen finden Sie unter [Zielverfolgungs-Skalierungsrichtlinien für Application Auto Scaling](#).

### Note

Die folgenden Beispiele sind für Unix, Linux und macOS formatiert. Ersetzen Sie unter Windows den umgekehrten Schrägstrich (\), das Unix-Fortsetzungszeichen, am Ende jeder Zeile durch ein Caret-Zeichen oder Zirkumflex (^).

## Einrichten der Zielverfolgung

Um die Zielverfolgung für einen Endpunkt einzurichten, verwenden Sie AWS CLI-Befehle, um ein skalierbares Ziel zu registrieren und dann eine Skalierungsrichtlinie zu erstellen. Das skalierbare Ziel definiert Inferenzeinheiten als die Ressource, die zur Anpassung der Endpunktbereitstellung verwendet wird, und die Skalierungsrichtlinie definiert die Metriken, die die automatische Skalierung der bereitgestellten Kapazität steuern.

## So richten Sie die Zielverfolgung ein

1. Registrieren eines skalierbaren Ziels In den folgenden Beispielen wird ein skalierbares Ziel registriert, um die Endpunktbereitstellung mit einer Mindestkapazität von 1 Inferenzeinheit und einer Höchstkapazität von 2 Inferenzeinheiten anzupassen.

Verwenden Sie für einen Endpunkt der Dokumentklassifizierung den folgenden AWS CLI-Befehl:

```
aws application-autoscaling register-scalable-target \
  --service-namespace comprehend \
  --resource-id arn:aws:comprehend:region:account-id:document-classifier-
endpoint/name \
  --scalable-dimension comprehend:document-classifier-
endpoint:DesiredInferenceUnits \
  --min-capacity 1 \
  --max-capacity 2
```

Verwenden Sie für einen Entity Recognizer-Endpunkt den folgenden AWS CLI-Befehl:

```
aws application-autoscaling register-scalable-target \
  --service-namespace comprehend \
  --resource-id arn:aws:comprehend:region:account-id:entity-recognizer-
endpoint/name \
  --scalable-dimension comprehend:entity-recognizer-
endpoint:DesiredInferenceUnits \
  --min-capacity 1 \
  --max-capacity 2
```

2. Verwenden Sie den folgenden AWS CLI-Befehl, um die Registrierung des skalierbaren Ziels zu überprüfen:

```
aws application-autoscaling describe-scalable-targets \
  --service-namespace comprehend \
  --resource-id endpoint ARN
```

3. Erstellen Sie eine Konfiguration für die Ziel-Nachverfolgung für die Skalierungsrichtlinie und speichern Sie die Konfiguration in einer Datei namens `config.json`. Im Folgenden finden Sie ein Beispiel für eine Zielverfolgungskonfiguration, die die Anzahl der Inferenzeinheiten

automatisch anpasst, sodass die genutzte Kapazität immer 70 % der bereitgestellten Kapazität ausmacht.

```
{
  "TargetValue": 70,
  "PredefinedMetricSpecification":
  {
    "PredefinedMetricType": "ComprehendInferenceUtilization"
  }
}
```

4. Erstellen Sie eine Skalierungsrichtlinie. In den folgenden Beispielen wird eine Skalierungsrichtlinie basierend auf der in der `config.json` Datei definierten Zielverfolgungskonfiguration erstellt.

Verwenden Sie für einen Endpunkt der Dokumentklassifizierung den folgenden AWS CLI-Befehl:

```
aws application-autoscaling put-scaling-policy \
  --service-namespace comprehend \
  --resource-id arn:aws:comprehend:region:account-id:document-classifier-
endpoint/name \
  --scalable-dimension comprehend:document-classifier-
endpoint:DesiredInferenceUnits \
  --policy-name TestPolicy \
  --policy-type TargetTrackingScaling \
  --target-tracking-scaling-policy-configuration file://config.json
```

Verwenden Sie für einen Entity Recognizer-Endpunkt den folgenden AWS CLI-Befehl:

```
aws application-autoscaling put-scaling-policy \
  --service-namespace comprehend \
  --resource-id arn:aws:comprehend:region:account-id:entity-recognizer-
endpoint/name \
  --scalable-dimension comprehend:entity-recognizer-
endpoint:DesiredInferenceUnits \
  --policy-name TestPolicy \
  --policy-type TargetTrackingScaling \
  --target-tracking-scaling-policy-configuration file://config.json
```

## Entfernen der Zielverfolgung

Um die Zielverfolgung für einen Endpunkt zu entfernen, verwenden Sie AWS CLI-Befehle, um die Skalierungsrichtlinie zu löschen und dann das skalierbare Ziel abzumelden.

So entfernen Sie die Zielverfolgung

1. Löschen Sie die Skalierungsrichtlinie. In den folgenden Beispielen wird eine angegebene Skalierungsrichtlinie gelöscht.

Verwenden Sie für einen Endpunkt der Dokumentklassifizierung den folgenden AWS CLI-Befehl:

```
aws application-autoscaling delete-scaling-policy \  
  --service-namespace comprehend \  
  --resource-id arn:aws:comprehend:region:account-id:document-classifier-  
endpoint/name \  
  --scalable-dimension comprehend:document-classifier-  
endpoint:DesiredInferenceUnits \  
  --policy-name TestPolicy \  

```

Verwenden Sie für einen Entity Recognizer-Endpunkt den folgenden AWS CLI-Befehl:

```
aws application-autoscaling delete-scaling-policy \  
  --service-namespace comprehend \  
  --resource-id arn:aws:comprehend:region:account-id:entity-recognizer-  
endpoint/name \  
  --scalable-dimension comprehend:entity-recognizer-  
endpoint:DesiredInferenceUnits \  
  --policy-name TestPolicy \  

```

2. Melden Sie das skalierbare Ziel ab. In den folgenden Beispielen wird die Registrierung eines angegebenen skalierbaren Ziels aufgehoben.

Verwenden Sie für einen Endpunkt der Dokumentklassifizierung den folgenden AWS CLI-Befehl:

```
aws application-autoscaling deregister-scalable-target \  
  --service-namespace comprehend \  
  --resource-id arn:aws:comprehend:region:account-id:document-classifier-  
endpoint/name \  

```

```
--scalable-dimension comprehend:document-classifier-  
endpoint:DesiredInferenceUnits
```

Verwenden Sie für einen Entity Recognizer-Endpoint den folgenden AWS CLI-Befehl:

```
aws application-autoscaling deregister-scalable-target \  
  --service-namespace comprehend \  
  --resource-id arn:aws:comprehend:region:account-id:entity-recognizer-  
endpoint/name \  
  --scalable-dimension comprehend:entity-recognizer-  
endpoint:DesiredInferenceUnits
```

## Geplante Skalierung

Mit der geplanten Skalierung können Sie die Endpunktbereitstellung an Ihre Kapazitätsanforderungen nach einem bestimmten Zeitplan anpassen. Die geplante Skalierung passt die Anzahl der Inferenzeinheiten automatisch an, um Nutzungsspitzen zu bestimmten Zeiten Rechnung zu tragen. Sie können die geplante Skalierung für Endpunkte zur Dokumentenklassifizierung und Endpunkte zur Entitätserkennung verwenden. Weitere Informationen zur geplanten Skalierung finden Sie unter [Geplante Skalierung für Application Auto Scaling](#).

### Note

Die folgenden Beispiele sind für Unix, Linux und macOS formatiert. Ersetzen Sie unter Windows den umgekehrten Schrägstrich (\), das Unix-Fortsetzungszeichen, am Ende jeder Zeile durch ein Caret-Zeichen oder Zirkumflex (^).

## Einrichten der geplanten Skalierung

Um die geplante Skalierung für einen Endpunkt einzurichten, verwenden Sie AWS CLI-Befehle, um ein skalierbares Ziel zu registrieren und dann eine geplante Aktion zu erstellen. Das skalierbare Ziel definiert Inferenzeinheiten als die Ressource, die zur Anpassung der Endpunktbereitstellung verwendet wird, und die geplante Aktion steuert die automatische Skalierung der bereitgestellten Kapazität zu bestimmten Zeiten.

## So richten Sie die geplante Skalierung ein

1. Registrieren eines skalierbaren Ziels In den folgenden Beispielen wird ein skalierbares Ziel registriert, um die Endpunktbereitstellung mit einer Mindestkapazität von 1 Inferenzeinheit und einer Höchstkapazität von 2 Inferenzeinheiten anzupassen.

Verwenden Sie für einen Endpunkt der Dokumentklassifizierung den folgenden AWS CLI-Befehl:

```
aws application-autoscaling register-scalable-target \  
  --service-namespace comprehend \  
  --resource-id arn:aws:comprehend:region:account-id:document-classifier-  
endpoint/name \  
  --scalable-dimension comprehend:document-classifier-  
endpoint:DesiredInferenceUnits \  
  --min-capacity 1 \  
  --max-capacity 2
```

Verwenden Sie für einen Entity Recognizer-Endpunkt den folgenden AWS CLI-Befehl:

```
aws application-autoscaling register-scalable-target \  
  --service-namespace comprehend \  
  --resource-id arn:aws:comprehend:region:account-id:entity-recognizer-  
endpoint/name \  
  --scalable-dimension comprehend:entity-recognizer-  
endpoint:DesiredInferenceUnits \  
  --min-capacity 1 \  
  --max-capacity 2
```

2. Erstellen Sie eine geplante Aktion. In den folgenden Beispielen wird eine geplante Aktion erstellt, um die bereitgestellte Kapazität jeden Tag um 12:00 Uhr UTC mit mindestens 2 Inferenzeinheiten und maximal 5 Inferenzeinheiten automatisch anzupassen. Weitere Informationen über chronologische Ausdrücke und geplante Skalierung finden Sie unter [Planen von Ausdrücken](#).

Verwenden Sie für einen Endpunkt der Dokumentklassifizierung den folgenden AWS CLI-Befehl:

```
aws application-autoscaling put-scheduled-action \  
  --service-namespace comprehend \  
  --resource-id arn:aws:comprehend:region:account-id:document-classifier-  
endpoint/name \  
  --schedule-expression
```

```
--scalable-dimension comprehend:document-classifier-
endpoint:DesiredInferenceUnits \
--scheduled-action-name TestScheduledAction \
--schedule "cron(0 12 * * ? *)" \
--scalable-target-action MinCapacity=2,MaxCapacity=5
```

Verwenden Sie für einen Entity Recognizer-Endpoint den folgenden AWS CLI-Befehl:

```
aws application-autoscaling put-scheduled-action \
--service-namespace comprehend \
--resource-id arn:aws:comprehend:region:account-id:entity-recognizer-
endpoint/name \
--scalable-dimension comprehend:entity-recognizer-
endpoint:DesiredInferenceUnits \
--scheduled-action-name TestScheduledAction \
--schedule "cron(0 12 * * ? *)" \
--scalable-target-action MinCapacity=2,MaxCapacity=5
```

## Entfernen der geplanten Skalierung

Um die geplante Skalierung für einen Endpoint zu entfernen, verwenden Sie AWS CLI-Befehle, um die geplante Aktion zu löschen und dann das skalierbare Ziel abzumelden.

So entfernen Sie die geplante Skalierung

1. Löschen Sie die geplante Aktion. In den folgenden Beispielen wird eine angegebene geplante Aktion gelöscht.

Verwenden Sie für einen Endpoint der Dokumentklassifizierung den folgenden AWS CLI-Befehl:

```
aws application-autoscaling delete-scheduled-action \
--service-namespace comprehend \
--resource-id arn:aws:comprehend:region:account-id:document-classifier-
endpoint/name \
--scalable-dimension comprehend:document-classifier-
endpoint:DesiredInferenceUnits \
--scheduled-action-name TestScheduledAction
```

Verwenden Sie für einen Entity Recognizer-Endpoint den folgenden AWS CLI-Befehl:

```
aws application-autoscaling delete-scheduled-action \  
  --service-namespace comprehend \  
  --resource-id arn:aws:comprehend:region:account-id:entity-recognizer-  
endpoint/name \  
  --scalable-dimension comprehend:entity-recognizer-  
endpoint:DesiredInferenceUnits \  
  --scheduled-action-name TestScheduledAction
```

2. Melden Sie das skalierbare Ziel ab. In den folgenden Beispielen wird die Registrierung eines angegebenen skalierbaren Ziels aufgehoben.

Verwenden Sie für einen Endpunkt der Dokumentklassifizierung den folgenden AWS CLI-Befehl:

```
aws application-autoscaling deregister-scalable-target \  
  --service-namespace comprehend \  
  --resource-id arn:aws:comprehend:region:account-id:document-classifier-  
endpoint/name \  
  --scalable-dimension comprehend:document-classifier-  
endpoint:DesiredInferenceUnits
```

Verwenden Sie für einen Entity Recognizer-Endpunkt den folgenden AWS CLI-Befehl:

```
aws application-autoscaling deregister-scalable-target \  
  --service-namespace comprehend \  
  --resource-id arn:aws:comprehend:region:account-id:entity-recognizer-  
endpoint/name \  
  --scalable-dimension comprehend:entity-recognizer-  
endpoint:DesiredInferenceUnits
```



# Markieren Ihrer -Ressourcen

Ein Tag ist ein Schlüssel-Wert-Paar, das Sie einer Amazon Comprehend-Ressource als Metadaten hinzufügen können. Sie können Tags für Analyseaufträge, benutzerdefinierte Klassifizierungsmodelle, benutzerdefinierte Entitätserkennungsmodelle und Endpunkte verwenden. Tags haben zwei Hauptfunktionen: die Organisation Ihrer Ressourcen und die Bereitstellung einer Tag-basierten Zugriffskontrolle.

Um Ihre Ressourcen mit Tags zu organisieren, können Sie den Tag-Schlüssel „Abteilung“ und die Tag-Werte „Vertrieb“ oder „Recht“ hinzufügen. Anschließend können Sie nach Ressourcen suchen und filtern, die für die Rechtsabteilung Ihres Unternehmens relevant sind.

Um eine Tag-basierte Zugriffskontrolle bereitzustellen, erstellen Sie IAM-Richtlinien mit Berechtigungen, die auf Tags basieren. Eine Richtlinie kann eine Operation basierend auf den in Ihrer Anforderung bereitgestellten Tags (request-tags) oder Tags, die der von Ihnen aufgerufenen Ressource zugeordnet sind (resource-tags), zulassen oder verweigern. Weitere Informationen zur Verwendung von Tags mit IAM finden Sie unter [Steuern des Zugriffs mithilfe von Tags](#) im IAM-Benutzerhandbuch.

Überlegungen zur Verwendung von Tags mit Amazon Comprehend:

- Sie können bis zu 50 Tags pro Ressource hinzufügen, und Tags können zum Zeitpunkt der Erstellung der Ressource oder rückwirkend hinzugefügt werden.
- Ein Tag-Schlüssel ist ein Pflichtfeld, aber ein Tag-Wert ist optional.
- Tags müssen nicht zwischen Ressourcen eindeutig sein, aber eine bestimmte Ressource darf keine doppelten Tag-Schlüssel haben.
- Bei Tag-Schlüsseln und -Werten muss die Groß- und Kleinschreibung beachtet werden.
- Ein Tag-Schlüssel kann maximal 127 Zeichen lang sein; ein Tag-Wert darf maximal 255 Zeichen lang sein.
- Das `aws :` Präfix ist für die AWS Verwendung reserviert. Sie können keine Tags hinzufügen, bearbeiten oder löschen, deren Schlüssel mit `beginntaws :` beginnt. Diese Tags werden nicht auf Ihr `tags-per-resource` Limit von 50 angerechnet.

**Note**

Wenn Sie Ihr Markierungsschema für mehrere AWS Services und Ressourcen verwenden möchten, denken Sie daran, dass andere Services möglicherweise unterschiedliche Anforderungen für zulässige Zeichen haben.

## Themen

- [Eine neue Ressource taggen](#)
- [Anzeigen, Bearbeiten und Löschen von Tags, die einer Ressource zugeordnet sind](#)

## Eine neue Ressource taggen

Sie können einem Analysejob, einem benutzerdefinierten Klassifizierungsmodell, einem benutzerdefinierten Entitätserkennungsmodell oder Endpunkten Tags hinzufügen.

1. [Melden Sie sich bei der Amazon Comprehend Comprehend-Konsole an AWS Management Console und öffnen Sie sie unter https://console.aws.amazon.com/comprehend/](https://console.aws.amazon.com/comprehend/)
2. Wählen Sie im linken Navigationsbereich die Ressource (Analysejob, Benutzerdefinierte Klassifizierung oder Benutzerdefinierte Entitätserkennung) aus, die Sie erstellen möchten.
3. Klicken Sie auf Job erstellen (oder Neues Modell erstellen). Dadurch gelangen Sie zur Hauptseite „Erstellen“ für Ihre Ressource. Unten auf dieser Seite siehst du das Feld „Tags — optional“.

▼ **Tags - optional** [Info](#)

A tag is a label that you can add to a resource as metadata to help you organize, search, or filter your data. Each tag consists of a key and an optional value.

Key	Value - optional	
<input type="text" value="Enter key"/>	<input type="text" value="Enter value"/>	<input type="button" value="Remove tag"/>
<input type="button" value="Add tag"/>		

Geben Sie einen Tag-Schlüssel und einen optionalen Tag-Wert ein. Wählen Sie Tag hinzufügen, um der Ressource ein weiteres Tag hinzuzufügen. Wiederholen Sie diesen Vorgang, bis alle Ihre Tags hinzugefügt sind. Beachten Sie, dass Tag-Schlüssel pro Ressource eindeutig sein müssen.

4. Wählen Sie die Schaltfläche „Auftrag erstellen“ oder „Auftrag erstellen“, um mit der Erstellung Ihrer Ressource fortzufahren.

Sie können Tags auch mit der AWS CLI hinzufügen. Dieses Beispiel zeigt, wie Sie mit dem [start-entities-detection-job](#) Befehl Tags hinzufügen.

```
aws comprehend start-entities-detection-job \  
--language-code "en" \  
--input-data-config "{\"S3Uri\": \"s3://test-input/TEST.csv\"}" \  
--output-data-config "{\"S3Uri\": \"s3://test-output\"}" \  
--data-access-role-arn arn:aws:iam:123456789012:role/test \  
--tags "[{\"Key\": \"color\", \"Value\": \"orange\"}]"
```

## Anzeigen, Bearbeiten und Löschen von Tags, die einer Ressource zugeordnet sind

Sie können Tags anzeigen, die einem Analyseauftrag, einem benutzerdefinierten Klassifizierungsmodell oder einem benutzerdefinierten Entitätserkennungsmodell zugeordnet sind.

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon Comprehend-Konsole unter <https://console.aws.amazon.com/comprehend/>
2. Wählen Sie die Ressource aus (Analyseauftrag, benutzerdefinierte Klassifizierung oder Erkennung benutzerdefinierter Entitäten), die die Datei mit den Tags enthält, die Sie anzeigen, ändern oder löschen möchten. Dadurch wird die Liste der vorhandenen Dateien für die von Ihnen ausgewählte Ressource angezeigt.

**Analysis jobs** [Info](#)

Analyze documents stored in Amazon S3 to find entities like events, phrases, primary language, sentiment, or personally identifiable information (PII).

**Analysis jobs (1)** Stop Duplicate Create job

Status: All < 1 > ⚙️

Name	Analysis type	Start	End
<input type="radio"/> <a href="#">my-comprehend-analysis-job</a>	Key phrases	10/22/2021, 10:43:57 AM	10/22/2021, 10:52:07 AM

- Klicken Sie auf den Namen der Datei (oder des Modells), deren Tags Sie anzeigen, ändern oder löschen möchten. Dadurch gelangen Sie zur Detailseite für diese Datei (oder dieses Modell). Scrollen Sie nach unten, bis Sie ein Feld Tags sehen. Hier können Sie alle Tags sehen, die mit Ihrer ausgewählten Datei (oder Ihrem Modell) verknüpft sind.

**Tags (2)** Manage tags

Key	Value
color	orange
type	PDF

Wählen Sie Tags verwalten aus, um Tags zu bearbeiten oder aus Ihrer Ressource zu entfernen.

- Klicken Sie auf den Text, den Sie ändern möchten, und bearbeiten Sie dann Ihr Tag. Sie können das Tag auch entfernen, indem Sie Tag entfernen auswählen. Um ein neues Tag hinzuzufügen, wählen Sie Tag hinzufügen und geben Sie dann den gewünschten Text in die leeren Felder ein.

**Tags** [Info](#)

A tag is a label that you can add to a resource as metadata to help you organize, search, or filter your data. Each tag consists of a key and an optional value.

Key	Value - optional	
<input type="text" value="color"/>	<input type="text" value="orange"/>	<span>Remove tag</span>
<input type="text" value="type"/>	<input type="text" value="PDF"/>	<span>Remove tag</span>

Add tag

Cancel Save

Wenn Sie mit dem Ändern Ihrer Tags fertig sind, wählen Sie Speichern aus.

# Codebeispiele für Amazon Comprehend mit SDKs AWS

Die folgenden Codebeispiele zeigen, wie Amazon Comprehend mit einem AWS Software Development Kit (SDK) verwendet wird.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Während Aktionen Ihnen zeigen, wie Sie einzelne Servicefunktionen aufrufen, können Sie Aktionen im Kontext der zugehörigen Szenarien und serviceübergreifenden Beispiele sehen.

Szenarien sind Codebeispiele, die Ihnen zeigen, wie Sie eine bestimmte Aufgabe ausführen können, indem Sie mehrere Funktionen innerhalb desselben Services aufrufen.

Serviceübergreifende Beispiele sind Beispielanwendungen, die über mehrere AWS-Services hinweg arbeiten.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Amazon Comprehend mit einem SDK verwenden AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

## Codebeispiele

- [Aktionen für Amazon Comprehend mithilfe von SDKs AWS](#)
  - [Verwendung CreateDocumentClassifier mit einem AWS SDK oder CLI](#)
  - [Verwendung DeleteDocumentClassifier mit einem AWS SDK oder CLI](#)
  - [Verwendung DescribeDocumentClassificationJob mit einem AWS SDK oder CLI](#)
  - [Verwendung DescribeDocumentClassifier mit einem AWS SDK oder CLI](#)
  - [Verwendung DescribeTopicsDetectionJob mit einem AWS SDK oder CLI](#)
  - [Verwendung DetectDominantLanguage mit einem AWS SDK oder CLI](#)
  - [Verwendung DetectEntities mit einem AWS SDK oder CLI](#)
  - [Verwendung DetectKeyPhrases mit einem AWS SDK oder CLI](#)
  - [Verwendung DetectPiiEntities mit einem AWS SDK oder CLI](#)
  - [Verwendung DetectSentiment mit einem AWS SDK oder CLI](#)
  - [Verwendung DetectSyntax mit einem AWS SDK oder CLI](#)
  - [Verwendung ListDocumentClassificationJobs mit einem AWS SDK oder CLI](#)
  - [Verwendung ListDocumentClassifiers mit einem AWS SDK oder CLI](#)
  - [Verwendung ListTopicsDetectionJobs mit einem AWS SDK oder CLI](#)

- [Verwendung StartDocumentClassificationJob mit einem AWS SDK oder CLI](#)
- [Verwendung StartTopicsDetectionJob mit einem AWS SDK oder CLI](#)
- [Szenarien für Amazon Comprehend mit SDKs AWS](#)
  - [Erkennen Sie Dokumentelemente mit Amazon Comprehend und einem SDK AWS](#)
  - [Führen Sie mithilfe eines SDK einen Amazon Comprehend-Themenmodellierungsjob für Beispieldaten aus AWS](#)
  - [Trainieren Sie einen benutzerdefinierten Amazon Comprehend-Klassifikator und klassifizieren Sie Dokumente mithilfe eines SDK AWS](#)
- [Serviceübergreifende Beispiele für Amazon Comprehend mit SDKs AWS](#)
  - [Erstellen einer Amazon-Transcribe-Streaming-App](#)
  - [Erstellen Sie einen Amazon Lex Lex-Chatbot, um die Besucher Ihrer Website anzusprechen](#)
  - [Erstellen Sie eine Webanwendung, die mithilfe von Amazon SQS Nachrichten sendet und abrufen](#)
  - [Erstellen einer Anwendung, die Kundenfeedback analysiert und Audio generiert](#)
  - [Erkennen Sie Entitäten in Text, der mithilfe eines AWS SDK aus einem Bild extrahiert wurde](#)

## Aktionen für Amazon Comprehend mithilfe von SDKs AWS

Die folgenden Codebeispiele zeigen, wie einzelne Amazon Comprehend Aktionen mit AWS SDKs ausgeführt werden. Diese Auszüge rufen die Amazon Comprehend API auf und sind Codeauszüge aus größeren Programmen, die im Kontext ausgeführt werden müssen. Jedes Beispiel enthält einen Link zu GitHub, wo Sie Anweisungen zum Einrichten und Ausführen des Codes finden.

Die folgenden Beispiele enthalten nur die am häufigsten verwendeten Aktionen. Eine vollständige Liste finden Sie in der [Amazon Comprehend API-Referenz](#).

### Beispiele

- [Verwendung CreateDocumentClassifier mit einem AWS SDK oder CLI](#)
- [Verwendung DeleteDocumentClassifier mit einem AWS SDK oder CLI](#)
- [Verwendung DescribeDocumentClassificationJob mit einem AWS SDK oder CLI](#)
- [Verwendung DescribeDocumentClassifier mit einem AWS SDK oder CLI](#)
- [Verwendung DescribeTopicsDetectionJob mit einem AWS SDK oder CLI](#)
- [Verwendung DetectDominantLanguage mit einem AWS SDK oder CLI](#)
- [Verwendung DetectEntities mit einem AWS SDK oder CLI](#)

- [Verwendung DetectKeyPhrases mit einem AWS SDK oder CLI](#)
- [Verwendung DetectPiiEntities mit einem AWS SDK oder CLI](#)
- [Verwendung DetectSentiment mit einem AWS SDK oder CLI](#)
- [Verwendung DetectSyntax mit einem AWS SDK oder CLI](#)
- [Verwendung ListDocumentClassificationJobs mit einem AWS SDK oder CLI](#)
- [Verwendung ListDocumentClassifiers mit einem AWS SDK oder CLI](#)
- [Verwendung ListTopicsDetectionJobs mit einem AWS SDK oder CLI](#)
- [Verwendung StartDocumentClassificationJob mit einem AWS SDK oder CLI](#)
- [Verwendung StartTopicsDetectionJob mit einem AWS SDK oder CLI](#)

## Verwendung **CreateDocumentClassifier** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird `CreateDocumentClassifier`.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- [Trainieren Sie einen benutzerdefinierten Klassifikator und klassifizieren Sie Dokumente](#)

### CLI

#### AWS CLI

Um einen Dokumentenklassifizierer zur Kategorisierung von Dokumenten zu erstellen

Mit dem folgenden `create-document-classifier` Beispiel wird der Trainingsprozess für ein Dokumentenklassifizierungsmodell gestartet. Die Trainingsdatendatei, `training.csv`, befindet sich am `--input-data-config` Tag. `training.csv` ist ein zweispaltiges Dokument, in dem die Bezeichnungen oder Klassifizierungen in der ersten Spalte und die Dokumente in der zweiten Spalte angegeben sind.

```
aws comprehend create-document-classifier \
  --document-classifier-name example-classifier \
  --data-access-arn arn:aws:comprehend:us-west-2:111122223333:pii-entities-
detection-job/123456abcdeb0e11022f22a11EXAMPLE \
  --input-data-config "S3Uri=s3://DOC-EXAMPLE-BUCKET/" \
```



```
--language-code en
```

Ausgabe:


```
{
  "DocumentClassifierArn": "arn:aws:comprehend:us-west-2:111122223333:document-
classifier/example-classifier"
}
```

Weitere Informationen finden Sie unter [Benutzerdefinierte Klassifizierung](#) im Amazon Comprehend Developer Guide.

- Einzelheiten zur API finden Sie unter [CreateDocumentClassifier AWS CLI Befehlsreferenz](#).

Java

SDK für Java 2.x

 Note

Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.comprehend.ComprehendClient;
import software.amazon.awssdk.services.comprehend.model.ComprehendException;
import
  software.amazon.awssdk.services.comprehend.model.CreateDocumentClassifierRequest;
import
  software.amazon.awssdk.services.comprehend.model.CreateDocumentClassifierResponse;
import
  software.amazon.awssdk.services.comprehend.model.DocumentClassifierInputDataConfig;

/**
 * Before running this code example, you can setup the necessary resources, such
 * as the CSV file and IAM Roles, by following this document:
 * https://aws.amazon.com/blogs/machine-learning/building-a-custom-classifier-
 * using-amazon-comprehend/
 *
 * Also, set up your development environment, including your credentials.
```

```
*
* For more information, see the following documentation topic:
*
* https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
*/
public class DocumentClassifierDemo {
    public static void main(String[] args) {
        final String usage = ""

            Usage:    <dataAccessRoleArn> <s3Uri> <documentClassifierName>

            Where:
                dataAccessRoleArn - The ARN value of the role used for this
operation.
                s3Uri - The Amazon S3 bucket that contains the CSV file.
                documentClassifierName - The name of the document classifier.
            "";

        if (args.length != 3) {
            System.out.println(usage);
            System.exit(1);
        }

        String dataAccessRoleArn = args[0];
        String s3Uri = args[1];
        String documentClassifierName = args[2];

        Region region = Region.US_EAST_1;
        ComprehendClient comClient = ComprehendClient.builder()
            .region(region)
            .build();

        createDocumentClassifier(comClient, dataAccessRoleArn, s3Uri,
documentClassifierName);
        comClient.close();
    }

    public static void createDocumentClassifier(ComprehendClient comClient,
String dataAccessRoleArn, String s3Uri,
        String documentClassifierName) {
        try {
            DocumentClassifierInputDataConfig config =
DocumentClassifierInputDataConfig.builder()

```

```

        .s3Uri(s3Uri)
        .build();

        CreateDocumentClassifierRequest createDocumentClassifierRequest =
CreateDocumentClassifierRequest.builder()
        .documentClassifierName(documentClassifierName)
        .dataAccessRoleArn(dataAccessRoleArn)
        .languageCode("en")
        .inputDataConfig(config)
        .build();

        CreateDocumentClassifierResponse createDocumentClassifierResult =
comClient
        .createDocumentClassifier(createDocumentClassifierRequest);
        String documentClassifierArn =
createDocumentClassifierResult.documentClassifierArn();
        System.out.println("Document Classifier ARN: " +
documentClassifierArn);

    } catch (ComprehendException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
}
}

```

- Einzelheiten zur API finden Sie [CreateDocumentClassifier](#) in der AWS SDK for Java 2.x API-Referenz.

## Python

### SDK für Python (Boto3)

#### Note

Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
class ComprehendClassifier:
```

```
"""Encapsulates an Amazon Comprehend custom classifier."""

def __init__(self, comprehend_client):
    """
    :param comprehend_client: A Boto3 Comprehend client.
    """
    self.comprehend_client = comprehend_client
    self.classifier_arn = None

def create(
    self,
    name,
    language_code,
    training_bucket,
    training_key,
    data_access_role_arn,
    mode,
):
    """
    Creates a custom classifier. After the classifier is created, it
    immediately
    starts training on the data found in the specified Amazon S3 bucket.
    Training
    can take 30 minutes or longer. The `describe_document_classifier`
    function
    can be used to get training status and returns a status of TRAINED when
    the
    classifier is ready to use.

    :param name: The name of the classifier.
    :param language_code: The language the classifier can operate on.
    :param training_bucket: The Amazon S3 bucket that contains the training
    data.
    :param training_key: The prefix used to find training data in the
    training
    bucket. If multiple objects have the same prefix,
    all
    of them are used.
    :param data_access_role_arn: The Amazon Resource Name (ARN) of a role
    that
    grants Comprehend permission to read from
    the
    training bucket.
    """
```

```
        :return: The ARN of the newly created classifier.
        """
        try:
            response = self.comprehend_client.create_document_classifier(
                DocumentClassifierName=name,
                LanguageCode=language_code,
                InputDataConfig={"S3Uri": f"s3://{training_bucket}/
{training_key}"},
                DataAccessRoleArn=data_access_role_arn,
                Mode=mode.value,
            )
            self.classifier_arn = response["DocumentClassifierArn"]
            logger.info("Started classifier creation. Arn is: %s.",
                self.classifier_arn)
        except ClientError:
            logger.exception("Couldn't create classifier %s.", name)
            raise
        else:
            return self.classifier_arn
```

- Einzelheiten zur API finden Sie [CreateDocumentClassifier](#) in AWS SDK for Python (Boto3) API Reference.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Amazon Comprehend mit einem SDK verwenden AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

## Verwendung **DeleteDocumentClassifier** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird `DeleteDocumentClassifier`.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- [Trainieren Sie einen benutzerdefinierten Klassifikator und klassifizieren Sie Dokumente](#)

## CLI

### AWS CLI

Um einen benutzerdefinierten Dokumentklassifikator zu löschen

Im folgenden `delete-document-classifier` Beispiel wird ein benutzerdefiniertes Dokumentklassifizierungsmodell gelöscht.

```
aws comprehend delete-document-classifier \  
  --document-classifier-arn arn:aws:comprehend:us-west-2:111122223333:document-  
classifier/example-classifier-1
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Managing Amazon Comprehend Endpoints](#) im Amazon Comprehend Developer Guide.

- Einzelheiten zur API finden Sie unter Befehlsreferenz. [DeleteDocumentClassifier](#) AWS CLI

## Python

### SDK für Python (Boto3)

#### Note

Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
class ComprehendClassifier:  
    """Encapsulates an Amazon Comprehend custom classifier."""  
  
    def __init__(self, comprehend_client):  
        """  
        :param comprehend_client: A Boto3 Comprehend client.  
        """  
        self.comprehend_client = comprehend_client  
        self.classifier_arn = None
```

```
def delete(self):
    """
    Deletes the classifier.
    """
    try:
        self.comprehend_client.delete_document_classifier(
            DocumentClassifierArn=self.classifier_arn
        )
        logger.info("Deleted classifier %s.", self.classifier_arn)
        self.classifier_arn = None
    except ClientError:
        logger.exception("Couldn't deleted classifier %s.",
            self.classifier_arn)
        raise
```

- Einzelheiten zur API finden Sie [DeleteDocumentClassifier](#) in AWS SDK for Python (Boto3) API Reference.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Amazon Comprehend mit einem SDK verwenden AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

## Verwendung **DescribeDocumentClassificationJob** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird `DescribeDocumentClassificationJob`.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- [Trainieren Sie einen benutzerdefinierten Klassifikator und klassifizieren Sie Dokumente](#)

### CLI

#### AWS CLI

Um einen Job zur Dokumentenklassifizierung zu beschreiben

Im folgenden `describe-document-classification-job` Beispiel werden die Eigenschaften eines asynchronen Dokumentenklassifizierungsauftrags abgerufen.

```
aws comprehend describe-document-classification-job \  
  --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

Ausgabe:

```
{  
  "DocumentClassificationJobProperties": {  
    "JobId": "123456abcdeb0e11022f22a11EXAMPLE",  
    "JobArn": "arn:aws:comprehend:us-west-2:111122223333:document-  
classification-job/123456abcdeb0e11022f22a11EXAMPLE",  
    "JobName": "exampleclassificationjob",  
    "JobStatus": "COMPLETED",  
    "SubmitTime": "2023-06-14T17:09:51.788000+00:00",  
    "EndTime": "2023-06-14T17:15:58.582000+00:00",  
    "DocumentClassifierArn": "arn:aws:comprehend:us-  
west-2:111122223333:document-classifier/mymodel/version/1",  
    "InputDataConfig": {  
      "S3Uri": "s3://DOC-EXAMPLE-BUCKET/jobdata/",  
      "InputFormat": "ONE_DOC_PER_LINE"  
    },  
    "OutputDataConfig": {  
      "S3Uri": "s3://DOC-EXAMPLE-DESTINATION-BUCKET/  
testfolder/111122223333-CLN-123456abcdeb0e11022f22a11EXAMPLE/output/  
output.tar.gz"  
    },  
    "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/  
AmazonComprehendServiceRole-servicerole"  
  }  
}
```

Weitere Informationen finden Sie unter [Benutzerdefinierte Klassifizierung](#) im Amazon Comprehend Developer Guide.

- Einzelheiten zur API finden Sie unter [DescribeDocumentClassificationJob AWS CLIBefehlsreferenz](#).



## Python

### SDK für Python (Boto3)

#### Note

Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
class ComprehendClassifier:
    """Encapsulates an Amazon Comprehend custom classifier."""

    def __init__(self, comprehend_client):
        """
        :param comprehend_client: A Boto3 Comprehend client.
        """
        self.comprehend_client = comprehend_client
        self.classifier_arn = None

    def describe_job(self, job_id):
        """
        Gets metadata about a classification job.

        :param job_id: The ID of the job to look up.
        :return: Metadata about the job.
        """
        try:
            response =
self.comprehend_client.describe_document_classification_job(
                JobId=job_id
            )
            job = response["DocumentClassificationJobProperties"]
            logger.info("Got classification job %s.", job["JobName"])
        except ClientError:
            logger.exception("Couldn't get classification job %s.", job_id)
            raise
        else:
            return job
```

- Einzelheiten zur API finden Sie [DescribeDocumentClassificationJob](#) in AWS SDK for Python (Boto3) API Reference.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Amazon Comprehend mit einem SDK verwenden AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

## Verwendung **DescribeDocumentClassifier** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird `DescribeDocumentClassifier`.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- [Trainieren Sie einen benutzerdefinierten Klassifikator und klassifizieren Sie Dokumente](#)

### CLI

#### AWS CLI

Um einen Dokumentenklassifikator zu beschreiben

Im folgenden `describe-document-classifier` Beispiel werden die Eigenschaften eines benutzerdefinierten Dokumentklassifizierungsmodells abgerufen.

```
aws comprehend describe-document-classifier \
  --document-classifier-arn arn:aws:comprehend:us-west-2:111122223333:document-
classifier/example-classifier-1
```

Ausgabe:

```
{
  "DocumentClassifierProperties": {
    "DocumentClassifierArn": "arn:aws:comprehend:us-
west-2:111122223333:document-classifier/example-classifier-1",
    "LanguageCode": "en",
    "Status": "TRAINED",
    "SubmitTime": "2023-06-13T19:04:15.735000+00:00",
    "EndTime": "2023-06-13T19:42:31.752000+00:00",
```

```
"TrainingStartTime": "2023-06-13T19:08:20.114000+00:00",
"TrainingEndTime": "2023-06-13T19:41:35.080000+00:00",
"InputDataConfig": {
  "DataFormat": "COMPREHEND_CSV",
  "S3Uri": "s3://DOC-EXAMPLE-BUCKET/trainingdata"
},
"OutputDataConfig": {},
"ClassifierMetadata": {
  "NumberOfLabels": 3,
  "NumberOfTrainedDocuments": 5016,
  "NumberOfTestDocuments": 557,
  "EvaluationMetrics": {
    "Accuracy": 0.9856,
    "Precision": 0.9919,
    "Recall": 0.9459,
    "F1Score": 0.9673,
    "MicroPrecision": 0.9856,
    "MicroRecall": 0.9856,
    "MicroF1Score": 0.9856,
    "HammingLoss": 0.0144
  }
},
"DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/AmazonComprehendServiceRole-example-role",
"Mode": "MULTI_CLASS"
}
```

Weitere Informationen finden Sie unter [Erstellen und Verwalten von benutzerdefinierten Modellen](#) im Amazon Comprehend Developer Guide.

- Einzelheiten zur API finden Sie unter [DescribeDocumentClassifier AWS CLI](#) Befehlsreferenz.

## Python

### SDK für Python (Boto3)

#### Note

Es gibt noch mehr dazu [GitHub](#). Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
class ComprehendClassifier:
    """Encapsulates an Amazon Comprehend custom classifier."""

    def __init__(self, comprehend_client):
        """
        :param comprehend_client: A Boto3 Comprehend client.
        """
        self.comprehend_client = comprehend_client
        self.classifier_arn = None

    def describe(self, classifier_arn=None):
        """
        Gets metadata about a custom classifier, including its current status.

        :param classifier_arn: The ARN of the classifier to look up.
        :return: Metadata about the classifier.
        """
        if classifier_arn is not None:
            self.classifier_arn = classifier_arn
        try:
            response = self.comprehend_client.describe_document_classifier(
                DocumentClassifierArn=self.classifier_arn
            )
            classifier = response["DocumentClassifierProperties"]
            logger.info("Got classifier %s.", self.classifier_arn)
        except ClientError:
            logger.exception("Couldn't get classifier %s.", self.classifier_arn)
            raise
        else:
            return classifier
```

- Einzelheiten zur API finden Sie [DescribeDocumentClassifier](#) in AWS SDK for Python (Boto3) API Reference.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Amazon Comprehend mit einem SDK verwenden AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

## Verwendung **DescribeTopicsDetectionJob** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird `DescribeTopicsDetectionJob`.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- [Führen Sie einen Job zur Themenmodellierung mit Beispieldaten aus](#)

### CLI

#### AWS CLI

Um einen Job zur Themenerkennung zu beschreiben

Im folgenden `describe-topics-detection-job` Beispiel werden die Eigenschaften eines asynchronen Themenerkennungsauftrags abgerufen.

```
aws comprehend describe-topics-detection-job \  
  --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

Ausgabe:

```
{  
  "TopicsDetectionJobProperties": {  
    "JobId": "123456abcdeb0e11022f22a11EXAMPLE",  
    "JobArn": "arn:aws:comprehend:us-west-2:111122223333:topics-detection-  
job/123456abcdeb0e11022f22a11EXAMPLE",  
    "JobName": "example_topics_detection",  
    "JobStatus": "IN_PROGRESS",  
    "SubmitTime": "2023-06-09T18:44:43.414000+00:00",  
    "InputDataConfig": {  
      "S3Uri": "s3://DOC-EXAMPLE-BUCKET",  
      "InputFormat": "ONE_DOC_PER_LINE"  
    },  
    "OutputDataConfig": {  
      "S3Uri": "s3://DOC-EXAMPLE-DESTINATION-BUCKET/  
testfolder/111122223333-TOPICS-123456abcdeb0e11022f22a11EXAMPLE/output/  
output.tar.gz"  
    },  
  },  
}
```

```
    "NumberOfTopics": 10,
    "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-examplerole"
  }
}
```

Weitere Informationen finden Sie unter [Async-Analyse für Amazon Comprehend Insights im Amazon Comprehend Developer Guide](#).

- Einzelheiten zur API finden Sie in der Befehlsreferenz. [DescribeTopicsDetectionJobAWS CLI](#)

## Python

### SDK für Python (Boto3)

#### Note

Es gibt noch mehr dazu [GitHub](#). Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
class ComprehendTopicModeler:
    """Encapsulates a Comprehend topic modeler."""

    def __init__(self, comprehend_client):
        """
        :param comprehend_client: A Boto3 Comprehend client.
        """
        self.comprehend_client = comprehend_client

    def describe_job(self, job_id):
        """
        Gets metadata about a topic modeling job.

        :param job_id: The ID of the job to look up.
        :return: Metadata about the job.
        """
        try:
            response = self.comprehend_client.describe_topics_detection_job(
                JobId=job_id
```

```
    )
    job = response["TopicsDetectionJobProperties"]
    logger.info("Got topic detection job %s.", job_id)
except ClientError:
    logger.exception("Couldn't get topic detection job %s.", job_id)
    raise
else:
    return job
```

- Einzelheiten zur API finden Sie [DescribeTopicsDetectionJob](#) in AWS SDK for Python (Boto3) API Reference.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Amazon Comprehend mit einem SDK verwenden AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

## Verwendung **DetectDominantLanguage** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird `DetectDominantLanguage`.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- [Dokumentelemente erkennen](#)

.NET

AWS SDK for .NET

### Note

Es gibt noch mehr dazu [GitHub](#). Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
using System;
using System.Threading.Tasks;
```

```
using Amazon.Comprehend;
using Amazon.Comprehend.Model;

/// <summary>
/// This example calls the Amazon Comprehend service to determine the
/// dominant language.
/// </summary>
public static class DetectDominantLanguage
{
    /// <summary>
    /// Calls Amazon Comprehend to determine the dominant language used in
    /// the sample text.
    /// </summary>
    public static async Task Main()
    {
        string text = "It is raining today in Seattle.";

        var comprehendClient = new
AmazonComprehendClient(Amazon.RegionEndpoint.USWest2);

        Console.WriteLine("Calling DetectDominantLanguage\n");
        var detectDominantLanguageRequest = new
DetectDominantLanguageRequest()
        {
            Text = text,
        };

        var detectDominantLanguageResponse = await
comprehendClient.DetectDominantLanguageAsync(detectDominantLanguageRequest);
        foreach (var dl in detectDominantLanguageResponse.Languages)
        {
            Console.WriteLine($"Language Code: {dl.LanguageCode}, Score:
{dl.Score}");
        }

        Console.WriteLine("Done");
    }
}
```

- Einzelheiten zur API finden Sie [DetectDominantLanguage](#) in der AWS SDK for .NET API-Referenz.



## CLI

### AWS CLI

Um die dominante Sprache des Eingabetextes zu erkennen

Im Folgenden wird der Eingabetext `detect-dominant-language` analysiert und die dominante Sprache identifiziert. Der Konfidenzwert des vortrainierten Modells wird ebenfalls ausgegeben.

```
aws comprehend detect-dominant-language \  
  --text "It is a beautiful day in Seattle."
```

Ausgabe:

```
{  
  "Languages": [  
    {  
      "LanguageCode": "en",  
      "Score": 0.9877256155014038  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Dominant Language](#) im Amazon Comprehend Developer Guide.

- Einzelheiten zur API finden Sie [DetectDominantLanguage](#) in der AWS CLI Befehlsreferenz.

## Java

### SDK für Java 2.x

#### Note

Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
import software.amazon.awssdk.regions.Region;  
import software.amazon.awssdk.services.comprehend.ComprehendClient;
```

```
import software.amazon.awssdk.services.comprehend.model.ComprehendException;
import
    software.amazon.awssdk.services.comprehend.model.DetectDominantLanguageRequest;
import
    software.amazon.awssdk.services.comprehend.model.DetectDominantLanguageResponse;
import software.amazon.awssdk.services.comprehend.model.DominantLanguage;
import java.util.List;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class DetectLanguage {
    public static void main(String[] args) {
        // Specify French text - "It is raining today in Seattle".
        String text = "Il pleut aujourd'hui à Seattle";
        Region region = Region.US_EAST_1;

        ComprehendClient comClient = ComprehendClient.builder()
            .region(region)
            .build();

        System.out.println("Calling DetectDominantLanguage");
        detectTheDominantLanguage(comClient, text);
        comClient.close();
    }

    public static void detectTheDominantLanguage(ComprehendClient comClient,
        String text) {
        try {
            DetectDominantLanguageRequest request =
                DetectDominantLanguageRequest.builder()
                    .text(text)
                    .build();

            DetectDominantLanguageResponse resp =
                comClient.detectDominantLanguage(request);
            List<DominantLanguage> allLanList = resp.languages();
            for (DominantLanguage lang : allLanList) {
```

```

        System.out.println("Language is " + lang.languageCode());
    }

    } catch (ComprehendException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
}

```

- Einzelheiten zur API finden Sie [DetectDominantLanguage](#) in der AWS SDK for Java 2.x API-Referenz.

## Python

### SDK für Python (Boto3)

#### Note

Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```

class ComprehendDetect:
    """Encapsulates Comprehend detection functions."""

    def __init__(self, comprehend_client):
        """
        :param comprehend_client: A Boto3 Comprehend client.
        """
        self.comprehend_client = comprehend_client

    def detect_languages(self, text):
        """
        Detects languages used in a document.

        :param text: The document to inspect.
        :return: The list of languages along with their confidence scores.
        """

```

```
try:
    response = self.comprehend_client.detect_dominant_language(Text=text)
    languages = response["Languages"]
    logger.info("Detected %s languages.", len(languages))
except ClientError:
    logger.exception("Couldn't detect languages.")
    raise
else:
    return languages
```

- Einzelheiten zur API finden Sie [DetectDominantLanguage](#) in AWS SDK for Python (Boto3) API Reference.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Amazon Comprehend mit einem SDK verwenden AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

## Verwendung **DetectEntities** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird `DetectEntities`.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- [Dokumentelemente erkennen](#)

### .NET

#### AWS SDK for .NET

##### Note

Es gibt noch mehr dazu [GitHub](#). Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
using System;
```

```
using System.Threading.Tasks;
using Amazon.Comprehend;
using Amazon.Comprehend.Model;

/// <summary>
/// This example shows how to use the AmazonComprehend service detect any
/// entities in submitted text.
/// </summary>
public static class DetectEntities
{
    /// <summary>
    /// The main method calls the DetectEntitiesAsync method to find any
    /// entities in the sample code.
    /// </summary>
    public static async Task Main()
    {
        string text = "It is raining today in Seattle";

        var comprehendClient = new AmazonComprehendClient();

        Console.WriteLine("Calling DetectEntities\n");
        var detectEntitiesRequest = new DetectEntitiesRequest()
        {
            Text = text,
            LanguageCode = "en",
        };
        var detectEntitiesResponse = await
comprehendClient.DetectEntitiesAsync(detectEntitiesRequest);

        foreach (var e in detectEntitiesResponse.Entities)
        {
            Console.WriteLine($"Text: {e.Text}, Type: {e.Type}, Score:
{e.Score}, BeginOffset: {e.BeginOffset}, EndOffset: {e.EndOffset}");
        }

        Console.WriteLine("Done");
    }
}
```

- Einzelheiten zur API finden Sie [DetectEntities](#) in der AWS SDK for .NET API-Referenz.

## CLI

## AWS CLI

Um benannte Entitäten im Eingabetext zu erkennen

Das folgende `detect-entities` Beispiel analysiert den Eingabetext und gibt die benannten Entitäten zurück. Der Konfidenzwert des vortrainierten Modells wird ebenfalls für jede Vorhersage ausgegeben.

```
aws comprehend detect-entities \  
  --language-code en \  
  --text "Hello Zhang Wei, I am John. Your AnyCompany Financial Services, LLC  
credit card \  
  account 1111-XXXX-1111-XXXX has a minimum payment of $24.53 that is due by  
July 31st. Based on your autopay settings, \  
  we will withdraw your payment on the due date from your bank account number  
XXXXXX1111 with the routing number XXXXX0000. \  
  Customer feedback for Sunshine Spa, 123 Main St, Anywhere. Send comments to  
Alice at AnySpa@example.com."
```

Ausgabe:

```
{  
  "Entities": [  
    {  
      "Score": 0.9994556307792664,  
      "Type": "PERSON",  
      "Text": "Zhang Wei",  
      "BeginOffset": 6,  
      "EndOffset": 15  
    },  
    {  
      "Score": 0.9981022477149963,  
      "Type": "PERSON",  
      "Text": "John",  
      "BeginOffset": 22,  
      "EndOffset": 26  
    },  
    {  
      "Score": 0.9986887574195862,  
      "Type": "ORGANIZATION",  
      "Text": "AnyCompany Financial Services, LLC",
```

```
    "BeginOffset": 33,
    "EndOffset": 67
  },
  {
    "Score": 0.9959119558334351,
    "Type": "OTHER",
    "Text": "1111-XXXX-1111-XXXX",
    "BeginOffset": 88,
    "EndOffset": 107
  },
  {
    "Score": 0.9708039164543152,
    "Type": "QUANTITY",
    "Text": ".53",
    "BeginOffset": 133,
    "EndOffset": 136
  },
  {
    "Score": 0.9987268447875977,
    "Type": "DATE",
    "Text": "July 31st",
    "BeginOffset": 152,
    "EndOffset": 161
  },
  {
    "Score": 0.9858865737915039,
    "Type": "OTHER",
    "Text": "XXXXXX1111",
    "BeginOffset": 271,
    "EndOffset": 281
  },
  {
    "Score": 0.9700471758842468,
    "Type": "OTHER",
    "Text": "XXXXX0000",
    "BeginOffset": 306,
    "EndOffset": 315
  },
  {
    "Score": 0.9591118693351746,
    "Type": "ORGANIZATION",
    "Text": "Sunshine Spa",
    "BeginOffset": 340,
    "EndOffset": 352
  }
```

```
    },
    {
      "Score": 0.9797496795654297,
      "Type": "LOCATION",
      "Text": "123 Main St",
      "BeginOffset": 354,
      "EndOffset": 365
    },
    {
      "Score": 0.994929313659668,
      "Type": "PERSON",
      "Text": "Alice",
      "BeginOffset": 394,
      "EndOffset": 399
    },
    {
      "Score": 0.9949769377708435,
      "Type": "OTHER",
      "Text": "AnySpa@example.com",
      "BeginOffset": 403,
      "EndOffset": 418
    }
  ]
}
```

Weitere Informationen finden Sie unter [Entitäten](#) im Amazon Comprehend Developer Guide.

- Einzelheiten zur API finden Sie [DetectEntities](#) in der AWS CLI Befehlsreferenz.

## Java

### SDK für Java 2.x

#### Note

Es gibt noch mehr dazu [GitHub](#). Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.comprehend.ComprehendClient;
import software.amazon.awssdk.services.comprehend.model.DetectEntitiesRequest;
```



```
import software.amazon.awssdk.services.comprehend.model.DetectEntitiesResponse;
import software.amazon.awssdk.services.comprehend.model.Entity;
import software.amazon.awssdk.services.comprehend.model.ComprehendException;
import java.util.List;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class DetectEntities {
    public static void main(String[] args) {
        String text = "Amazon.com, Inc. is located in Seattle, WA and was founded
        July 5th, 1994 by Jeff Bezos, allowing customers to buy everything from books to
        blenders. Seattle is north of Portland and south of Vancouver, BC. Other notable
        Seattle - based companies are Starbucks and Boeing.";
        Region region = Region.US_EAST_1;
        ComprehendClient comClient = ComprehendClient.builder()
            .region(region)
            .build();

        System.out.println("Calling DetectEntities");
        detectAllEntities(comClient, text);
        comClient.close();
    }

    public static void detectAllEntities(ComprehendClient comClient, String text)
    {
        try {
            DetectEntitiesRequest detectEntitiesRequest =
            DetectEntitiesRequest.builder()
                .text(text)
                .languageCode("en")
                .build();

            DetectEntitiesResponse detectEntitiesResult =
            comClient.detectEntities(detectEntitiesRequest);
            List<Entity> entList = detectEntitiesResult.entities();
            for (Entity entity : entList) {
                System.out.println("Entity text is " + entity.text());
            }
        }
    }
}
```

```

        }

    } catch (ComprehendException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
}

```

- Einzelheiten zur API finden Sie [DetectEntities](#) in der AWS SDK for Java 2.x API-Referenz.

## Python

### SDK für Python (Boto3)

#### Note

Es gibt noch mehr dazu [GitHub](#). Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```

class ComprehendDetect:
    """Encapsulates Comprehend detection functions."""

    def __init__(self, comprehend_client):
        """
        :param comprehend_client: A Boto3 Comprehend client.
        """
        self.comprehend_client = comprehend_client

    def detect_entities(self, text, language_code):
        """
        Detects entities in a document. Entities can be things like people and
        places
        or other common terms.

        :param text: The document to inspect.
        :param language_code: The language of the document.
        :return: The list of entities along with their confidence scores.
        """

```

```
try:
    response = self.comprehend_client.detect_entities(
        Text=text, LanguageCode=language_code
    )
    entities = response["Entities"]
    logger.info("Detected %s entities.", len(entities))
except ClientError:
    logger.exception("Couldn't detect entities.")
    raise
else:
    return entities
```

- Einzelheiten zur API finden Sie [DetectEntities](#) in AWS SDK for Python (Boto3) API Reference.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Amazon Comprehend mit einem SDK verwenden AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

## Verwendung **DetectKeyPhrases** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird `DetectKeyPhrases`.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- [Dokumentelemente erkennen](#)

.NET

AWS SDK for .NET

### Note

Es gibt noch mehr dazu [GitHub](#). Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
using System;
using System.Threading.Tasks;
using Amazon.Comprehend;
using Amazon.Comprehend.Model;

/// <summary>
/// This example shows how to use the Amazon Comprehend service to
/// search text for key phrases.
/// </summary>
public static class DetectKeyPhrase
{
    /// <summary>
    /// This method calls the Amazon Comprehend method DetectKeyPhrasesAsync
    /// to detect any key phrases in the sample text.
    /// </summary>
    public static async Task Main()
    {
        string text = "It is raining today in Seattle";

        var comprehendClient = new
AmazonComprehendClient(Amazon.RegionEndpoint.USWest2);

        // Call DetectKeyPhrases API
        Console.WriteLine("Calling DetectKeyPhrases");
        var detectKeyPhrasesRequest = new DetectKeyPhrasesRequest()
        {
            Text = text,
            LanguageCode = "en",
        };
        var detectKeyPhrasesResponse = await
comprehendClient.DetectKeyPhrasesAsync(detectKeyPhrasesRequest);
        foreach (var kp in detectKeyPhrasesResponse.KeyPhrases)
        {
            Console.WriteLine($"Text: {kp.Text}, Score: {kp.Score},
BeginOffset: {kp.BeginOffset}, EndOffset: {kp.EndOffset}");
        }

        Console.WriteLine("Done");
    }
}
```

- Einzelheiten zur API finden Sie [DetectKeyPhrases](#) in der AWS SDK for .NET API-Referenz.

## CLI

### AWS CLI

Um Schlüsselphrasen im Eingabetext zu erkennen

Das folgende `detect-key-phrases` Beispiel analysiert den Eingabetext und identifiziert die wichtigsten Nominalphrasen. Der Konfidenzwert des vortrainierten Modells wird ebenfalls für jede Vorhersage ausgegeben.

```
aws comprehend detect-key-phrases \  
  --language-code en \  
  --text "Hello Zhang Wei, I am John. Your AnyCompany Financial Services, LLC  
credit card \  
  account 1111-XXXX-1111-XXXX has a minimum payment of $24.53 that is due  
by July 31st. Based on your autopay settings, \  
  we will withdraw your payment on the due date from your bank account  
number XXXXXX1111 with the routing number XXXXX0000. \  
  Customer feedback for Sunshine Spa, 123 Main St, Anywhere. Send comments  
to Alice at AnySpa@example.com."
```

Ausgabe:

```
{  
  "KeyPhrases": [  
    {  
      "Score": 0.8996376395225525,  
      "Text": "Zhang Wei",  
      "BeginOffset": 6,  
      "EndOffset": 15  
    },  
    {  
      "Score": 0.9992469549179077,  
      "Text": "John",  
      "BeginOffset": 22,  
      "EndOffset": 26  
    },  
    {  
      "Score": 0.988385021686554,  
      "Text": "Your AnyCompany Financial Services",
```

```
    "BeginOffset": 28,
    "EndOffset": 62
  },
  {
    "Score": 0.8740853071212769,
    "Text": "LLC credit card account 1111-XXXX-1111-XXXX",
    "BeginOffset": 64,
    "EndOffset": 107
  },
  {
    "Score": 0.9999437928199768,
    "Text": "a minimum payment",
    "BeginOffset": 112,
    "EndOffset": 129
  },
  {
    "Score": 0.9998900890350342,
    "Text": ".53",
    "BeginOffset": 133,
    "EndOffset": 136
  },
  {
    "Score": 0.9979453086853027,
    "Text": "July 31st",
    "BeginOffset": 152,
    "EndOffset": 161
  },
  {
    "Score": 0.9983011484146118,
    "Text": "your autopay settings",
    "BeginOffset": 172,
    "EndOffset": 193
  },
  {
    "Score": 0.9996572136878967,
    "Text": "your payment",
    "BeginOffset": 211,
    "EndOffset": 223
  },
  {
    "Score": 0.9995037317276001,
    "Text": "the due date",
    "BeginOffset": 227,
    "EndOffset": 239
  }
```

```
    },
    {
      "Score": 0.9702621698379517,
      "Text": "your bank account number XXXXXX1111",
      "BeginOffset": 245,
      "EndOffset": 280
    },
    {
      "Score": 0.9179925918579102,
      "Text": "the routing number XXXXX0000.Customer feedback",
      "BeginOffset": 286,
      "EndOffset": 332
    },
    {
      "Score": 0.9978160858154297,
      "Text": "Sunshine Spa",
      "BeginOffset": 337,
      "EndOffset": 349
    },
    {
      "Score": 0.9706913232803345,
      "Text": "123 Main St",
      "BeginOffset": 351,
      "EndOffset": 362
    },
    {
      "Score": 0.9941995143890381,
      "Text": "comments",
      "BeginOffset": 379,
      "EndOffset": 387
    },
    {
      "Score": 0.9759287238121033,
      "Text": "Alice",
      "BeginOffset": 391,
      "EndOffset": 396
    },
    {
      "Score": 0.8376792669296265,
      "Text": "AnySpa@example.com",
      "BeginOffset": 400,
      "EndOffset": 415
    }
  ]
```

```
}
```

Weitere Informationen finden Sie unter [Schlüsselbegriffe](#) im Amazon Comprehend Developer Guide.

- Einzelheiten zur API finden Sie [DetectKeyPhrases](#) in der AWS CLI Befehlsreferenz.

## Java

### SDK für Java 2.x

#### Note

Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.comprehend.ComprehendClient;
import software.amazon.awssdk.services.comprehend.model.DetectKeyPhrasesRequest;
import software.amazon.awssdk.services.comprehend.model.DetectKeyPhrasesResponse;
import software.amazon.awssdk.services.comprehend.model.KeyPhrase;
import software.amazon.awssdk.services.comprehend.model.ComprehendException;
import java.util.List;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
 * started.html
 */
public class DetectKeyPhrases {
    public static void main(String[] args) {
        String text = "Amazon.com, Inc. is located in Seattle, WA and was founded
        July 5th, 1994 by Jeff Bezos, allowing customers to buy everything from books to
        blenders. Seattle is north of Portland and south of Vancouver, BC. Other notable
        Seattle - based companies are Starbucks and Boeing.";
        Region region = Region.US_EAST_1;
        ComprehendClient comClient = ComprehendClient.builder()
```



```
        .region(region)
        .build();

        System.out.println("Calling DetectKeyPhrases");
        detectAllKeyPhrases(comClient, text);
        comClient.close();
    }

    public static void detectAllKeyPhrases(ComprehendClient comClient, String
text) {
        try {
            DetectKeyPhrasesRequest detectKeyPhrasesRequest =
DetectKeyPhrasesRequest.builder()
                .text(text)
                .languageCode("en")
                .build();

            DetectKeyPhrasesResponse detectKeyPhrasesResult =
comClient.detectKeyPhrases(detectKeyPhrasesRequest);
            List<KeyPhrase> phraseList = detectKeyPhrasesResult.keyPhrases();
            for (KeyPhrase keyPhrase : phraseList) {
                System.out.println("Key phrase text is " + keyPhrase.text());
            }

        } catch (ComprehendException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Einzelheiten zur API finden Sie [DetectKeyPhrases](#) in der AWS SDK for Java 2.x API-Referenz.

## Python

## SDK für Python (Boto3)

 Note

Es gibt noch mehr dazu [GitHub](#). Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
class ComprehendDetect:
    """Encapsulates Comprehend detection functions."""

    def __init__(self, comprehend_client):
        """
        :param comprehend_client: A Boto3 Comprehend client.
        """
        self.comprehend_client = comprehend_client

    def detect_key_phrases(self, text, language_code):
        """
        Detects key phrases in a document. A key phrase is typically a noun and
its
        modifiers.

        :param text: The document to inspect.
        :param language_code: The language of the document.
        :return: The list of key phrases along with their confidence scores.
        """
        try:
            response = self.comprehend_client.detect_key_phrases(
                Text=text, LanguageCode=language_code
            )
            phrases = response["KeyPhrases"]
            logger.info("Detected %s phrases.", len(phrases))
        except ClientError:
            logger.exception("Couldn't detect phrases.")
            raise
        else:
            return phrases
```

- Einzelheiten zur API finden Sie [DetectKeyPhrases](#) in AWS SDK for Python (Boto3) API Reference.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Amazon Comprehend mit einem SDK verwenden AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

## Verwendung **DetectPiiEntities** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird `DetectPiiEntities`.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- [Dokumentelemente erkennen](#)

.NET

AWS SDK for .NET

### Note

Es gibt noch mehr dazu [GitHub](#). Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
using System;
using System.Threading.Tasks;
using Amazon.Comprehend;
using Amazon.Comprehend.Model;

/// <summary>
/// This example shows how to use the Amazon Comprehend service to find
/// personally identifiable information (PII) within text submitted to the
/// DetectPiiEntitiesAsync method.
/// </summary>
public class DetectingPII
```

```
{
    /// <summary>
    /// This method calls the DetectPiiEntitiesAsync method to locate any
    /// personally identifiable information within the supplied text.
    /// </summary>
    public static async Task Main()
    {
        var comprehendClient = new AmazonComprehendClient();
        var text = @"Hello Paul Santos. The latest statement for your
                    credit card account 1111-0000-1111-0000 was
                    mailed to 123 Any Street, Seattle, WA 98109.";

        var request = new DetectPiiEntitiesRequest
        {
            Text = text,
            LanguageCode = "EN",
        };

        var response = await
comprehendClient.DetectPiiEntitiesAsync(request);

        if (response.Entities.Count > 0)
        {
            foreach (var entity in response.Entities)
            {
                var entityValue = text.Substring(entity.BeginOffset,
entity.EndOffset - entity.BeginOffset);
                Console.WriteLine($"{entity.Type}: {entityValue}");
            }
        }
    }
}
```

- Einzelheiten zur API finden Sie [DetectPiiEntities](#) in der AWS SDK for .NET API-Referenz.

## CLI

### AWS CLI

Um PII-Entitäten im Eingabetext zu erkennen

Das folgende `detect-pii-entities` Beispiel analysiert den Eingabetext und identifiziert Entitäten, die personenbezogene Daten (PII) enthalten. Der Konfidenzwert des vortrainierten Modells wird ebenfalls für jede Vorhersage ausgegeben.

```
aws comprehend detect-pii-entities \  
  --language-code en \  
  --text "Hello Zhang Wei, I am John. Your AnyCompany Financial Services, LLC  
credit card \  
  account 1111-XXXX-1111-XXXX has a minimum payment of $24.53 that is due  
by July 31st. Based on your autopay settings, \  
  we will withdraw your payment on the due date from your bank account  
number XXXXXX1111 with the routing number XXXXX0000. \  
  Customer feedback for Sunshine Spa, 123 Main St, Anywhere. Send comments  
to Alice at AnySpa@example.com."
```

Ausgabe:

```
{  
  "Entities": [  
    {  
      "Score": 0.9998322129249573,  
      "Type": "NAME",  
      "BeginOffset": 6,  
      "EndOffset": 15  
    },  
    {  
      "Score": 0.9998878240585327,  
      "Type": "NAME",  
      "BeginOffset": 22,  
      "EndOffset": 26  
    },  
    {  
      "Score": 0.9994089603424072,  
      "Type": "CREDIT_DEBIT_NUMBER",  
      "BeginOffset": 88,  
      "EndOffset": 107  
    },  
    {  
      "Score": 0.9999760985374451,  
      "Type": "DATE_TIME",  
      "BeginOffset": 152,  
      "EndOffset": 161  
    },  
  ],  
}
```


```
{
  "Score": 0.9999449253082275,
  "Type": "BANK_ACCOUNT_NUMBER",
  "BeginOffset": 271,
  "EndOffset": 281
},
{
  "Score": 0.9999847412109375,
  "Type": "BANK_ROUTING",
  "BeginOffset": 306,
  "EndOffset": 315
},
{
  "Score": 0.999925434589386,
  "Type": "ADDRESS",
  "BeginOffset": 354,
  "EndOffset": 365
},
{
  "Score": 0.9989161491394043,
  "Type": "NAME",
  "BeginOffset": 394,
  "EndOffset": 399
},
{
  "Score": 0.9994171857833862,
  "Type": "EMAIL",
  "BeginOffset": 403,
  "EndOffset": 418
}
]
```

Weitere Informationen finden Sie unter [Persönlich Identifizierbare Informationen \(PII\)](#) im Amazon Comprehend Developer Guide.

- Einzelheiten zur API finden Sie unter Befehlsreferenz [DetectPiiEntities](#).AWS CLI

## Python

## SDK für Python (Boto3)

 Note

Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
class ComprehendDetect:
    """Encapsulates Comprehend detection functions."""

    def __init__(self, comprehend_client):
        """
        :param comprehend_client: A Boto3 Comprehend client.
        """
        self.comprehend_client = comprehend_client

    def detect_pii(self, text, language_code):
        """
        Detects personally identifiable information (PII) in a document. PII can
        be
        things like names, account numbers, or addresses.

        :param text: The document to inspect.
        :param language_code: The language of the document.
        :return: The list of PII entities along with their confidence scores.
        """
        try:
            response = self.comprehend_client.detect_pii_entities(
                Text=text, LanguageCode=language_code
            )
            entities = response["Entities"]
            logger.info("Detected %s PII entities.", len(entities))
        except ClientError:
            logger.exception("Couldn't detect PII entities.")
            raise
        else:
            return entities
```

- Einzelheiten zur API finden Sie [DetectPiiEntities](#) in AWS SDK for Python (Boto3) API Reference.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Amazon Comprehend mit einem SDK verwenden AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

## Verwendung **DetectSentiment** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird `DetectSentiment`.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- [Dokumentelemente erkennen](#)

.NET

AWS SDK for .NET

### Note

Es gibt noch mehr dazu [GitHub](#). Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
using System;
using System.Threading.Tasks;
using Amazon.Comprehend;
using Amazon.Comprehend.Model;

/// <summary>
/// This example shows how to detect the overall sentiment of the supplied
/// text using the Amazon Comprehend service.
/// </summary>
public static class DetectSentiment
{
```



```
/// <summary>
/// This method calls the DetectSentimentAsync method to analyze the
/// supplied text and determine the overall sentiment.
/// </summary>
public static async Task Main()
{
    string text = "It is raining today in Seattle";

    var comprehendClient = new
AmazonComprehendClient(Amazon.RegionEndpoint.USWest2);

    // Call DetectKeyPhrases API
    Console.WriteLine("Calling DetectSentiment");
    var detectSentimentRequest = new DetectSentimentRequest()
    {
        Text = text,
        LanguageCode = "en",
    };
    var detectSentimentResponse = await
comprehendClient.DetectSentimentAsync(detectSentimentRequest);
    Console.WriteLine($"Sentiment: {detectSentimentResponse.Sentiment}");
    Console.WriteLine("Done");
}
}
```

- Einzelheiten zur API finden Sie [DetectSentiment](#) in der AWS SDK for .NET API-Referenz.

## CLI

### AWS CLI

Um die Stimmung eines eingegebenen Textes zu erkennen

Das folgende `detect-sentiment` Beispiel analysiert den Eingabetext und gibt einen Rückschluss auf die vorherrschende Stimmung (POSITIVE, NEUTRALMIXED, oder) zurück.  
NEGATIVE

```
aws comprehend detect-sentiment \
  --language-code en \
  --text "It is a beautiful day in Seattle"
```

**Ausgabe:**

```
{
  "Sentiment": "POSITIVE",
  "SentimentScore": {
    "Positive": 0.9976957440376282,
    "Negative": 9.653854067437351e-05,
    "Neutral": 0.002169104292988777,
    "Mixed": 3.857641786453314e-05
  }
}
```

Weitere Informationen finden Sie unter [Sentiment](#) im Amazon Comprehend Developer Guide.

- Einzelheiten zur API finden Sie [DetectSentiment](#) in AWS CLI der Befehlsreferenz.

**Java****SDK für Java 2.x****Note**

Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.comprehend.ComprehendClient;
import software.amazon.awssdk.services.comprehend.model.ComprehendException;
import software.amazon.awssdk.services.comprehend.model.DetectSentimentRequest;
import software.amazon.awssdk.services.comprehend.model.DetectSentimentResponse;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
 * started.html
 */
```

```
public class DetectSentiment {
    public static void main(String[] args) {
        String text = "Amazon.com, Inc. is located in Seattle, WA and was founded
        July 5th, 1994 by Jeff Bezos, allowing customers to buy everything from books to
        blenders. Seattle is north of Portland and south of Vancouver, BC. Other notable
        Seattle - based companies are Starbucks and Boeing.";
        Region region = Region.US_EAST_1;
        ComprehendClient comClient = ComprehendClient.builder()
            .region(region)
            .build();

        System.out.println("Calling DetectSentiment");
        detectSentiments(comClient, text);
        comClient.close();
    }

    public static void detectSentiments(ComprehendClient comClient, String text)
    {
        try {
            DetectSentimentRequest detectSentimentRequest =
            DetectSentimentRequest.builder()
                .text(text)
                .languageCode("en")
                .build();


            DetectSentimentResponse detectSentimentResult =
            comClient.detectSentiment(detectSentimentRequest);
            System.out.println("The Neutral value is " +
            detectSentimentResult.sentimentScore().neutral());

        } catch (ComprehendException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Einzelheiten zur API finden Sie [DetectSentiment](#) in der AWS SDK for Java 2.x API-Referenz.

## Python

## SDK für Python (Boto3)

 Note

Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
class ComprehendDetect:
    """Encapsulates Comprehend detection functions."""

    def __init__(self, comprehend_client):
        """
        :param comprehend_client: A Boto3 Comprehend client.
        """
        self.comprehend_client = comprehend_client

    def detect_sentiment(self, text, language_code):
        """
        Detects the overall sentiment expressed in a document. Sentiment can
        be positive, negative, neutral, or a mixture.

        :param text: The document to inspect.
        :param language_code: The language of the document.
        :return: The sentiments along with their confidence scores.
        """
        try:
            response = self.comprehend_client.detect_sentiment(
                Text=text, LanguageCode=language_code
            )
            logger.info("Detected primary sentiment %s.", response["Sentiment"])
        except ClientError:
            logger.exception("Couldn't detect sentiment.")
            raise
        else:
            return response
```

- Einzelheiten zur API finden Sie [DetectSentiment](#) in AWS SDK for Python (Boto3) API Reference.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Amazon Comprehend mit einem SDK verwenden AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

## Verwendung **DetectSyntax** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird `DetectSyntax`.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- [Dokumentelemente erkennen](#)

### .NET

#### AWS SDK for .NET

##### Note

Es gibt noch mehr dazu [GitHub](#). Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
using System;
using System.Threading.Tasks;
using Amazon.Comprehend;
using Amazon.Comprehend.Model;

/// <summary>
/// This example shows how to use Amazon Comprehend to detect syntax
/// elements by calling the DetectSyntaxAsync method.
/// </summary>
public class DetectingSyntax
{
    /// <summary>
    /// This method calls DetectSynaxAsync to identify the syntax elements
    /// in the sample text.
```

```
/// </summary>
public static async Task Main()
{
    string text = "It is raining today in Seattle";

    var comprehendClient = new AmazonComprehendClient();

    // Call DetectSyntax API
    Console.WriteLine("Calling DetectSyntaxAsync\n");
    var detectSyntaxRequest = new DetectSyntaxRequest()
    {
        Text = text,
        LanguageCode = "en",
    };
    DetectSyntaxResponse detectSyntaxResponse = await
comprehendClient.DetectSyntaxAsync(detectSyntaxRequest);
    foreach (SyntaxToken s in detectSyntaxResponse.SyntaxTokens)
    {
        Console.WriteLine($"Text: {s.Text}, PartOfSpeech:
{s.PartOfSpeech.Tag}, BeginOffset: {s.BeginOffset}, EndOffset: {s.EndOffset}");
    }

    Console.WriteLine("Done");
}
}
```

- Einzelheiten zur API finden Sie [DetectSyntax](#) in der AWS SDK for .NET API-Referenz.

## CLI

### AWS CLI

Um die Wortarten in einem Eingabetext zu erkennen

Im folgenden `detect-syntax` Beispiel wird die Syntax des Eingabetextes analysiert und die verschiedenen Wortarten zurückgegeben. Der Konfidenzwert des vortrainierten Modells wird ebenfalls für jede Vorhersage ausgegeben.

```
aws comprehend detect-syntax \  
--language-code en \  

```

```
--text "It is a beautiful day in Seattle."
```

Ausgabe:

```
{
  "SyntaxTokens": [
    {
      "TokenId": 1,
      "Text": "It",
      "BeginOffset": 0,
      "EndOffset": 2,
      "PartOfSpeech": {
        "Tag": "PRON",
        "Score": 0.9999740719795227
      }
    },
    {
      "TokenId": 2,
      "Text": "is",
      "BeginOffset": 3,
      "EndOffset": 5,
      "PartOfSpeech": {
        "Tag": "VERB",
        "Score": 0.999901294708252
      }
    },
    {
      "TokenId": 3,
      "Text": "a",
      "BeginOffset": 6,
      "EndOffset": 7,
      "PartOfSpeech": {
        "Tag": "DET",
        "Score": 0.9999938607215881
      }
    },
    {
      "TokenId": 4,
      "Text": "beautiful",
      "BeginOffset": 8,
      "EndOffset": 17,
      "PartOfSpeech": {
        "Tag": "ADJ",

```

```
        "Score": 0.9987351894378662
      }
    },
    {
      "TokenId": 5,
      "Text": "day",
      "BeginOffset": 18,
      "EndOffset": 21,
      "PartOfSpeech": {
        "Tag": "NOUN",
        "Score": 0.9999796748161316
      }
    },
    {
      "TokenId": 6,
      "Text": "in",
      "BeginOffset": 22,
      "EndOffset": 24,
      "PartOfSpeech": {
        "Tag": "ADP",
        "Score": 0.9998047947883606
      }
    },
    {
      "TokenId": 7,
      "Text": "Seattle",
      "BeginOffset": 25,
      "EndOffset": 32,
      "PartOfSpeech": {
        "Tag": "PROPN",
        "Score": 0.9940530061721802
      }
    }
  ]
}
```


Weitere Informationen finden Sie unter [Syntaxanalyse](#) im Amazon Comprehend Developer Guide.

- Einzelheiten zur API finden Sie [DetectSyntax](#) in der AWS CLI Befehlsreferenz.



## Java

## SDK für Java 2.x

 Note

Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.comprehend.ComprehendClient;
import software.amazon.awssdk.services.comprehend.model.ComprehendException;
import software.amazon.awssdk.services.comprehend.model.DetectSyntaxRequest;
import software.amazon.awssdk.services.comprehend.model.DetectSyntaxResponse;
import software.amazon.awssdk.services.comprehend.model.SyntaxToken;
import java.util.List;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
 * started.html
 */
public class DetectSyntax {
    public static void main(String[] args) {
        String text = "Amazon.com, Inc. is located in Seattle, WA and was founded
        July 5th, 1994 by Jeff Bezos, allowing customers to buy everything from books to
        blenders. Seattle is north of Portland and south of Vancouver, BC. Other notable
        Seattle - based companies are Starbucks and Boeing.";
        Region region = Region.US_EAST_1;
        ComprehendClient comClient = ComprehendClient.builder()
            .region(region)
            .build();

        System.out.println("Calling DetectSyntax");
        detectAllSyntax(comClient, text);
        comClient.close();
    }
}
```

```
public static void detectAllSyntax(ComprehendClient comClient, String text) {
    try {
        DetectSyntaxRequest detectSyntaxRequest =
DetectSyntaxRequest.builder()
            .text(text)
            .languageCode("en")
            .build();

        DetectSyntaxResponse detectSyntaxResult =
comClient.detectSyntax(detectSyntaxRequest);
        List<SyntaxToken> syntaxTokens = detectSyntaxResult.syntaxTokens();
        for (SyntaxToken token : syntaxTokens) {
            System.out.println("Language is " + token.text());
            System.out.println("Part of speech is " +
token.partOfSpeech().tagAsString());
        }

    } catch (ComprehendException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
}
```

- Einzelheiten zur API finden Sie [DetectSyntax](#) in der AWS SDK for Java 2.x API-Referenz.

## Python

### SDK für Python (Boto3)

#### Note

Es gibt noch mehr dazu [GitHub](#). Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
class ComprehendDetect:
    """Encapsulates Comprehend detection functions."""

    def __init__(self, comprehend_client):
```

```
"""
:param comprehend_client: A Boto3 Comprehend client.
"""
self.comprehend_client = comprehend_client

def detect_syntax(self, text, language_code):
    """
    Detects syntactical elements of a document. Syntax tokens are portions of
    text along with their use as parts of speech, such as nouns, verbs, and
    interjections.

    :param text: The document to inspect.
    :param language_code: The language of the document.
    :return: The list of syntax tokens along with their confidence scores.
    """
    try:
        response = self.comprehend_client.detect_syntax(
            Text=text, LanguageCode=language_code
        )
        tokens = response["SyntaxTokens"]
        logger.info("Detected %s syntax tokens.", len(tokens))
    except ClientError:
        logger.exception("Couldn't detect syntax.")
        raise
    else:
        return tokens
```

- Einzelheiten zur API finden Sie [DetectSyntax](#) in AWS SDK for Python (Boto3) API Reference.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Amazon Comprehend mit einem SDK verwenden AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

## Verwendung `ListDocumentClassificationJobs` mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird `ListDocumentClassificationJobs`.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- [Trainieren Sie einen benutzerdefinierten Klassifikator und klassifizieren Sie Dokumente](#)

### CLI

#### AWS CLI

Zur Liste aller Jobs zur Dokumentenklassifizierung

Das folgende `list-document-classification-jobs` Beispiel listet alle Aufträge zur Dokumentenklassifizierung auf.

```
aws comprehend list-document-classification-jobs
```

Ausgabe:

```
{
  "DocumentClassificationJobPropertiesList": [
    {
      "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
      "JobArn": "arn:aws:comprehend:us-west-2:1234567890101:document-classification-job/123456abcdeb0e11022f22a11EXAMPLE",
      "JobName": "exampleclassificationjob",
      "JobStatus": "COMPLETED",
      "SubmitTime": "2023-06-14T17:09:51.788000+00:00",
      "EndTime": "2023-06-14T17:15:58.582000+00:00",
      "DocumentClassifierArn": "arn:aws:comprehend:us-west-2:1234567890101:document-classifier/mymodel/version/12",
      "InputDataConfig": {
        "S3Uri": "s3://DOC-EXAMPLE-BUCKET/jobdata/",
        "InputFormat": "ONE_DOC_PER_LINE"
      },
      "OutputDataConfig": {
```

```

        "S3Uri": "s3://DOC-EXAMPLE-DESTINATION-BUCKET/
thefolder/1234567890101-CLN-e758dd56b824aa717ceab551f11749fb/output/
output.tar.gz"
    },
    "DataAccessRoleArn": "arn:aws:iam::1234567890101:role/service-role/
AmazonComprehendServiceRole-example-role"
  },
  {
    "JobId": "123456abcdeb0e11022f22a1EXAMPLE2",
    "JobArn": "arn:aws:comprehend:us-west-2:1234567890101:document-
classification-job/123456abcdeb0e11022f22a1EXAMPLE2",
    "JobName": "exampleclassificationjob2",
    "JobStatus": "COMPLETED",
    "SubmitTime": "2023-06-14T17:22:39.829000+00:00",
    "EndTime": "2023-06-14T17:28:46.107000+00:00",
    "DocumentClassifierArn": "arn:aws:comprehend:us-
west-2:1234567890101:document-classifier/mymodel/version/12",
    "InputDataConfig": {
      "S3Uri": "s3://DOC-EXAMPLE-BUCKET/jobdata/",
      "InputFormat": "ONE_DOC_PER_LINE"
    },
    "OutputDataConfig": {
      "S3Uri": "s3://DOC-EXAMPLE-DESTINATION-BUCKET/
thefolder/1234567890101-CLN-123456abcdeb0e11022f22a1EXAMPLE2/output/
output.tar.gz"
    },
    "DataAccessRoleArn": "arn:aws:iam::1234567890101:role/service-role/
AmazonComprehendServiceRole-example-role"
  }
]
}

```

Weitere Informationen finden Sie unter [Benutzerdefinierte Klassifizierung](#) im Amazon Comprehend Developer Guide.

- Einzelheiten zur API finden Sie unter [ListDocumentClassificationJobs AWS CLIBefehlsreferenz](#).

## Python

### SDK für Python (Boto3)

#### Note

Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
class ComprehendClassifier:
    """Encapsulates an Amazon Comprehend custom classifier."""

    def __init__(self, comprehend_client):
        """
        :param comprehend_client: A Boto3 Comprehend client.
        """
        self.comprehend_client = comprehend_client
        self.classifier_arn = None

    def list_jobs(self):
        """
        Lists the classification jobs for the current account.

        :return: The list of jobs.
        """
        try:
            response = self.comprehend_client.list_document_classification_jobs()
            jobs = response["DocumentClassificationJobPropertiesList"]
            logger.info("Got %s document classification jobs.", len(jobs))
        except ClientError:
            logger.exception(
                "Couldn't get document classification jobs.",
            )
            raise
        else:
            return jobs
```

- Einzelheiten zur API finden Sie [ListDocumentClassificationJobs](#) in AWS SDK for Python (Boto3) API Reference.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Amazon Comprehend mit einem SDK verwenden AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

## Verwendung **ListDocumentClassifiers** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird `ListDocumentClassifiers`.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- [Trainieren Sie einen benutzerdefinierten Klassifikator und klassifizieren Sie Dokumente](#)

### CLI

#### AWS CLI

Zur Liste aller Dokumentenklassifikatoren

Das folgende `list-document-classifiers` Beispiel listet alle trainierten und trainierten Dokumentenklassifizierungsmodelle auf.

```
aws comprehend list-document-classifiers
```

Ausgabe:

```
{
  "DocumentClassifierPropertiesList": [
    {
      "DocumentClassifierArn": "arn:aws:comprehend:us-west-2:111122223333:document-classifier/exampleclassifier1",
      "LanguageCode": "en",
      "Status": "TRAINED",
      "SubmitTime": "2023-06-13T19:04:15.735000+00:00",
      "EndTime": "2023-06-13T19:42:31.752000+00:00",
      "TrainingStartTime": "2023-06-13T19:08:20.114000+00:00",
      "TrainingEndTime": "2023-06-13T19:41:35.080000+00:00",
      "InputDataConfig": {
```

```

        "DataFormat": "COMPREHEND_CSV",
        "S3Uri": "s3://DOC-EXAMPLE-BUCKET/trainingdata"
    },
    "OutputDataConfig": {},
    "ClassifierMetadata": {
        "NumberOfLabels": 3,
        "NumberOfTrainedDocuments": 5016,
        "NumberOfTestDocuments": 557,
        "EvaluationMetrics": {
            "Accuracy": 0.9856,
            "Precision": 0.9919,
            "Recall": 0.9459,
            "F1Score": 0.9673,
            "MicroPrecision": 0.9856,
            "MicroRecall": 0.9856,
            "MicroF1Score": 0.9856,
            "HammingLoss": 0.0144
        }
    },
    "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/AmazonComprehendServiceRole-testorle",
    "Mode": "MULTI_CLASS"
},
{
    "DocumentClassifierArn": "arn:aws:comprehend:us-west-2:111122223333:document-classifier/exampleclassifier2",
    "LanguageCode": "en",
    "Status": "TRAINING",
    "SubmitTime": "2023-06-13T21:20:28.690000+00:00",
    "InputDataConfig": {
        "DataFormat": "COMPREHEND_CSV",
        "S3Uri": "s3://DOC-EXAMPLE-BUCKET/trainingdata"
    },
    "OutputDataConfig": {},
    "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/AmazonComprehendServiceRole-testorle",
    "Mode": "MULTI_CLASS"
}
]
}

```

Weitere Informationen finden Sie unter [Erstellen und Verwalten von benutzerdefinierten Modellen](#) im Amazon Comprehend Developer Guide.



- Einzelheiten zur API finden Sie unter [ListDocumentClassifiers AWS CLI Befehlsreferenz](#).

## Python

### SDK für Python (Boto3)

#### Note

Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
class ComprehendClassifier:
    """Encapsulates an Amazon Comprehend custom classifier."""

    def __init__(self, comprehend_client):
        """
        :param comprehend_client: A Boto3 Comprehend client.
        """
        self.comprehend_client = comprehend_client
        self.classifier_arn = None

    def list(self):
        """
        Lists custom classifiers for the current account.

        :return: The list of classifiers.
        """
        try:
            response = self.comprehend_client.list_document_classifiers()
            classifiers = response["DocumentClassifierPropertiesList"]
            logger.info("Got %s classifiers.", len(classifiers))
        except ClientError:
            logger.exception(
                "Couldn't get classifiers.",
            )
            raise
        else:
            return classifiers
```

- Einzelheiten zur API finden Sie [ListDocumentClassifiers](#) in AWS SDK for Python (Boto3) API Reference.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Amazon Comprehend mit einem SDK verwenden AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

## Verwendung **ListTopicsDetectionJobs** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird `ListTopicsDetectionJobs`.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- [Führen Sie einen Job zur Themenmodellierung mit Beispieldaten aus](#)

### CLI

#### AWS CLI

Um alle Jobs zur Themenerkennung aufzulisten

Das folgende `list-topics-detection-jobs` Beispiel listet alle laufenden und abgeschlossenen asynchronen Themenerkennungsaufträge auf.

```
aws comprehend list-topics-detection-jobs
```

Ausgabe:

```
{
  "TopicsDetectionJobPropertiesList": [
    {
      "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
      "JobArn": "arn:aws:comprehend:us-west-2:111122223333:topics-detection-job/123456abcdeb0e11022f22a11EXAMPLE",
      "JobName": "topic-analysis-1",
      "JobStatus": "IN_PROGRESS",
      "SubmitTime": "2023-06-09T18:40:35.384000+00:00",
      "EndTime": "2023-06-09T18:46:41.936000+00:00",
    }
  ]
}
```

```
    "InputDataConfig": {
      "S3Uri": "s3://DOC-EXAMPLE-BUCKET",
      "InputFormat": "ONE_DOC_PER_LINE"
    },
    "OutputDataConfig": {
      "S3Uri": "s3://DOC-EXAMPLE-DESTINATION-BUCKET/
thefolder/111122223333-TOPICS-123456abcdeb0e11022f22a11EXAMPLE/output/
output.tar.gz"
    },
    "NumberOfTopics": 10,
    "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role"
  },
  {
    "JobId": "123456abcdeb0e11022f22a1EXAMPLE2",
    "JobArn": "arn:aws:comprehend:us-west-2:111122223333:topics-
detection-job/123456abcdeb0e11022f22a1EXAMPLE2",
    "JobName": "topic-analysis-2",
    "JobStatus": "COMPLETED",
    "SubmitTime": "2023-06-09T18:44:43.414000+00:00",
    "EndTime": "2023-06-09T18:50:50.872000+00:00",
    "InputDataConfig": {
      "S3Uri": "s3://DOC-EXAMPLE-BUCKET",
      "InputFormat": "ONE_DOC_PER_LINE"
    },
    "OutputDataConfig": {
      "S3Uri": "s3://DOC-EXAMPLE-DESTINATION-BUCKET/
thefolder/111122223333-TOPICS-123456abcdeb0e11022f22a1EXAMPLE2/output/
output.tar.gz"
    },
    "NumberOfTopics": 10,
    "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role"
  },
  {
    "JobId": "123456abcdeb0e11022f22a1EXAMPLE3",
    "JobArn": "arn:aws:comprehend:us-west-2:111122223333:topics-
detection-job/123456abcdeb0e11022f22a1EXAMPLE3",
    "JobName": "topic-analysis-2",
    "JobStatus": "IN_PROGRESS",
    "SubmitTime": "2023-06-09T18:50:56.737000+00:00",
    "InputDataConfig": {
      "S3Uri": "s3://DOC-EXAMPLE-BUCKET",
      "InputFormat": "ONE_DOC_PER_LINE"
    }
  }
}
```

```

        },
        "OutputDataConfig": {
            "S3Uri": "s3://DOC-EXAMPLE-DESTINATION-BUCKET/
thefolder/111122223333-TOPICS-123456abcdeb0e11022f22a1EXAMPLE3/output/
output.tar.gz"
        },
        "NumberOfTopics": 10,
        "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role"
    }
]
}

```

Weitere Informationen finden Sie unter [Async-Analyse für Amazon Comprehend Insights im Amazon Comprehend Developer Guide](#).

- Einzelheiten zur API finden Sie in der Befehlsreferenz. [ListTopicsDetectionJobs](#) AWS CLI

## Python

### SDK für Python (Boto3)

#### Note

Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```

class ComprehendTopicModeler:
    """Encapsulates a Comprehend topic modeler."""

    def __init__(self, comprehend_client):
        """
        :param comprehend_client: A Boto3 Comprehend client.
        """
        self.comprehend_client = comprehend_client

    def list_jobs(self):
        """
        Lists topic modeling jobs for the current account.

```

```
:return: The list of jobs.
"""
try:
    response = self.comprehend_client.list_topics_detection_jobs()
    jobs = response["TopicsDetectionJobPropertiesList"]
    logger.info("Got %s topic detection jobs.", len(jobs))
except ClientError:
    logger.exception("Couldn't get topic detection jobs.")
    raise
else:
    return jobs
```

- Einzelheiten zur API finden Sie [ListTopicsDetectionJobs](#) in AWS SDK for Python (Boto3) API Reference.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Amazon Comprehend mit einem SDK verwenden AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

## Verwendung **StartDocumentClassificationJob** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird `StartDocumentClassificationJob`.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- [Trainieren Sie einen benutzerdefinierten Klassifikator und klassifizieren Sie Dokumente](#)

### CLI

#### AWS CLI

Um den Job zur Dokumentenklassifizierung zu starten

Im folgenden `start-document-classification-job` Beispiel wird ein Auftrag zur Dokumentenklassifizierung mit einem benutzerdefinierten Modell für alle Dateien an der durch das `--input-data-config` Tag angegebenen

Adresse gestartet. In diesem Beispiel enthält der S3-Eingabe-Bucket `SampleSMStext1.txt`, `SampleSMStext2.txt`, und `SampleSMStext3.txt`. Das Modell wurde zuvor anhand von Dokumentenklassifizierungen von Spam- und Nicht-Spam-SMS-Nachrichten bzw. „betrügerischen“ SMS-Nachrichten trainiert. Wenn der Job abgeschlossen ist, `output.tar.gz` wird er an der durch das `--output-data-config` Tag angegebenen Stelle platziert. `output.tar.gz` enthält `predictions.jsonl`, in dem die Klassifikation der einzelnen Dokumente aufgeführt ist. Die Json-Ausgabe wird in einer Zeile pro Datei gedruckt, ist hier aber aus Gründen der Lesbarkeit formatiert.

```
aws comprehend start-document-classification-job \  
  --job-name exampleclassificationjob \  
  --input-data-config "S3Uri=s3://DOC-EXAMPLE-BUCKET-INPUT/jobdata/" \  
  --output-data-config "S3Uri=s3://DOC-EXAMPLE-DESTINATION-BUCKET/testfolder/" \  
  \  
  --data-access-role-arn arn:aws:iam::111122223333:role/service-role/  
AmazonComprehendServiceRole-example-role \  
  --document-classifier-arn arn:aws:comprehend:us-west-2:111122223333:document-  
classifier/mymodel/version/12
```

Inhalt von `SampleSMStext1.txt`:

```
"CONGRATULATIONS! TXT 2155550100 to win $5000"
```

Inhalt von `SampleSMStext2.txt`:

```
"Hi, when do you want me to pick you up from practice?"
```

Inhalt von `SampleSMStext3.txt`:

```
"Plz send bank account # to 2155550100 to claim prize!!"
```

Ausgabe:

```
{  
  "JobId": "e758dd56b824aa717ceab551fEXAMPLE",  
  "JobArn": "arn:aws:comprehend:us-west-2:111122223333:document-classification-  
job/e758dd56b824aa717ceab551fEXAMPLE",  
  "JobStatus": "SUBMITTED"  
}
```

## Inhalt von predictions.jsonl:

```
{"File": "SampleSMSText1.txt", "Line": "0", "Classes": [{"Name": "spam", "Score": 0.9999}, {"Name": "ham", "Score": 0.0001}]}
{"File": "SampleSMStext2.txt", "Line": "0", "Classes": [{"Name": "ham", "Score": 0.9994}, {"Name": "spam", "Score": 0.0006}]}
{"File": "SampleSMSText3.txt", "Line": "0", "Classes": [{"Name": "spam", "Score": 0.9999}, {"Name": "ham", "Score": 0.0001}]}
```

Weitere Informationen finden Sie unter [Benutzerdefinierte Klassifizierung](#) im Amazon Comprehend Developer Guide.

- Einzelheiten zur API finden Sie unter [StartDocumentClassificationJob AWS CLIBefehlsreferenz](#).

## Python

### SDK für Python (Boto3)

#### Note

Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
class ComprehendClassifier:
    """Encapsulates an Amazon Comprehend custom classifier."""

    def __init__(self, comprehend_client):
        """
        :param comprehend_client: A Boto3 Comprehend client.
        """
        self.comprehend_client = comprehend_client
        self.classifier_arn = None

    def start_job(
        self,
        job_name,
        input_bucket,
        input_key,
```

```

        input_format,
        output_bucket,
        output_key,
        data_access_role_arn,
    ):
        """
        Starts a classification job. The classifier must be trained or the job
        will fail. Input is read from the specified Amazon S3 input bucket and
        written to the specified output bucket. Output data is stored in a tar
        archive compressed in gzip format. The job runs asynchronously, so you
        can
        call `describe_document_classification_job` to get job status until it
        returns a status of SUCCEEDED.

        :param job_name: The name of the job.
        :param input_bucket: The Amazon S3 bucket that contains input data.
        :param input_key: The prefix used to find input data in the input
            bucket. If multiple objects have the same prefix, all
            of them are used.
        :param input_format: The format of the input data, either one document
        per
            file or one document per line.
        :param output_bucket: The Amazon S3 bucket where output data is written.
        :param output_key: The prefix prepended to the output data.
        :param data_access_role_arn: The Amazon Resource Name (ARN) of a role
        that
            grants Comprehend permission to read from
        the
            input bucket and write to the output bucket.
        :return: Information about the job, including the job ID.
        """
        try:
            response = self.comprehend_client.start_document_classification_job(
                DocumentClassifierArn=self.classifier_arn,
                JobName=job_name,
                InputDataConfig={
                    "S3Uri": f"s3://{input_bucket}/{input_key}",
                    "InputFormat": input_format.value,
                },
                OutputDataConfig={"S3Uri": f"s3://{output_bucket}/{output_key}"},
                DataAccessRoleArn=data_access_role_arn,
            )
            logger.info(

```



```
        "Document classification job %s is %s.", job_name,
response["JobStatus"]
    )
    except ClientError:
        logger.exception("Couldn't start classification job %s.", job_name)
        raise
    else:
        return response
```

- Einzelheiten zur API finden Sie [StartDocumentClassificationJob](#) in AWS SDK for Python (Boto3) API Reference.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Amazon Comprehend mit einem SDK verwenden AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

## Verwendung **StartTopicsDetectionJob** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird `StartTopicsDetectionJob`.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- [Führen Sie einen Job zur Themenmodellierung mit Beispieldaten aus](#)

.NET

AWS SDK for .NET

### Note

Es gibt noch mehr dazu [GitHub](#). Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
using System;
using System.Threading.Tasks;
using Amazon.Comprehend;
```

```
using Amazon.Comprehend.Model;

/// <summary>
/// This example scans the documents in an Amazon Simple Storage Service
/// (Amazon S3) bucket and analyzes it for topics. The results are stored
/// in another bucket and then the resulting job properties are displayed
/// on the screen. This example was created using the AWS SDK for .NET
/// version 3.7 and .NET Core version 5.0.
/// </summary>
public static class TopicModeling
{
    /// <summary>
    /// This method calls a topic detection job by calling the Amazon
    /// Comprehend StartTopicsDetectionJobRequest.
    /// </summary>
    public static async Task Main()
    {
        var comprehendClient = new AmazonComprehendClient();

        string inputS3Uri = "s3://input bucket/input path";
        InputFormat inputDocFormat = InputFormat.ONE_DOC_PER_FILE;
        string outputS3Uri = "s3://output bucket/output path";
        string dataAccessRoleArn = "arn:aws:iam::account ID:role/data access
role";

        int numberOfTopics = 10;

        var startTopicsDetectionJobRequest = new
StartTopicsDetectionJobRequest()
        {
            InputDataConfig = new InputDataConfig()
            {
                S3Uri = inputS3Uri,
                InputFormat = inputDocFormat,
            },
            OutputDataConfig = new OutputDataConfig()
            {
                S3Uri = outputS3Uri,
            },
            DataAccessRoleArn = dataAccessRoleArn,
            NumberOfTopics = numberOfTopics,
        };

        var startTopicsDetectionJobResponse = await
comprehendClient.StartTopicsDetectionJobAsync(startTopicsDetectionJobRequest);
```

```
        var jobId = startTopicsDetectionJobResponse.JobId;
        Console.WriteLine("JobId: " + jobId);

        var describeTopicsDetectionJobRequest = new
DescribeTopicsDetectionJobRequest()
        {
            JobId = jobId,
        };

        var describeTopicsDetectionJobResponse = await
comprehendClient.DescribeTopicsDetectionJobAsync(describeTopicsDetectionJobRequest);

PrintJobProperties(describeTopicsDetectionJobResponse.TopicsDetectionJobProperties);

        var listTopicsDetectionJobsResponse = await
comprehendClient.ListTopicsDetectionJobsAsync(new
ListTopicsDetectionJobsRequest());
        foreach (var props in
listTopicsDetectionJobsResponse.TopicsDetectionJobPropertiesList)
        {
            PrintJobProperties(props);
        }
    }

    /// <summary>
    /// This method is a helper method that displays the job properties
    /// from the call to StartTopicsDetectionJobRequest.
    /// </summary>
    /// <param name="props">A list of properties from the call to
    /// StartTopicsDetectionJobRequest.</param>
    private static void PrintJobProperties(TopicsDetectionJobProperties
props)
    {
        Console.WriteLine($"JobId: {props.JobId}, JobName: {props.JobName},
JobStatus: {props.JobStatus}");
        Console.WriteLine($"NumberOfTopics:
{props.NumberOfTopics}\nInputS3Uri: {props.InputDataConfig.S3Uri}");
        Console.WriteLine($"InputFormat: {props.InputDataConfig.InputFormat},
OutputS3Uri: {props.OutputDataConfig.S3Uri}");
    }
}
```

- Einzelheiten zur API finden Sie [StartTopicsDetectionJob](#) in der AWS SDK for .NET API-Referenz.

## CLI

### AWS CLI

Um einen Analysejob zur Themenerkennung zu starten

Im folgenden `start-topics-detection-job` Beispiel wird ein asynchroner Auftrag zur Themenerkennung für alle Dateien gestartet, die sich an der durch das `--input-data-config` Tag angegebenen Adresse befinden. Wenn der Job abgeschlossen ist, wird der Ordner, `output`, an dem durch das `--output-data-config` Tag angegebenen Speicherort platziert. `output` enthält `topic-terms.csv` und `doc-topics.csv`. Die erste Ausgabedatei, `topic-terms.csv`, ist eine Liste von Themen in der Sammlung. Für jedes Thema enthält die Liste standardmäßig die wichtigsten Begriffe, sortiert nach Themen, entsprechend ihrer Gewichtung. In der zweiten Datei werden die Dokumente aufgeführt `doc-topics.csv`, die einem Thema zugeordnet sind, sowie der Anteil des Dokuments, der sich mit dem Thema befasst.

```
aws comprehend start-topics-detection-job \
  --job-name example_topics_detection_job \
  --language-code en \
  --input-data-config "S3Uri=s3://DOC-EXAMPLE-BUCKET/" \
  --output-data-config "S3Uri=s3://DOC-EXAMPLE-DESTINATION-BUCKET/testfolder/" \
  --data-access-role-arn arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role \
  --language-code en
```

Ausgabe:

```
{
  "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
  "JobArn": "arn:aws:comprehend:us-west-2:111122223333:key-phrases-detection-
job/123456abcdeb0e11022f22a11EXAMPLE",
  "JobStatus": "SUBMITTED"
}
```

Weitere Informationen finden Sie unter [Topic Modeling](#) im Amazon Comprehend Developer Guide.

- Einzelheiten zur API finden Sie [StartTopicsDetectionJob](#) in der AWS CLI Befehlsreferenz.

## Python

### SDK für Python (Boto3)

#### Note

Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
class ComprehendTopicModeler:
    """Encapsulates a Comprehend topic modeler."""

    def __init__(self, comprehend_client):
        """
        :param comprehend_client: A Boto3 Comprehend client.
        """
        self.comprehend_client = comprehend_client

    def start_job(
        self,
        job_name,
        input_bucket,
        input_key,
        input_format,
        output_bucket,
        output_key,
        data_access_role_arn,
    ):
        """
        Starts a topic modeling job. Input is read from the specified Amazon S3
        input bucket and written to the specified output bucket. Output data is
        stored
        in a tar archive compressed in gzip format. The job runs asynchronously,
        so you
        can call `describe_topics_detection_job` to get job status until it

```

```

returns a status of SUCCEEDED.

:param job_name: The name of the job.
:param input_bucket: An Amazon S3 bucket that contains job input.
:param input_key: The prefix used to find input data in the input
all
                    bucket. If multiple objects have the same prefix,
                    of them are used.
:param input_format: The format of the input data, either one document
per
                    file or one document per line.
:param output_bucket: The Amazon S3 bucket where output data is written.
:param output_key: The prefix prepended to the output data.
:param data_access_role_arn: The Amazon Resource Name (ARN) of a role
that
                    grants Comprehend permission to read from
the
                    input bucket and write to the output bucket.
:return: Information about the job, including the job ID.
"""
try:
    response = self.comprehend_client.start_topics_detection_job(
        JobName=job_name,
        DataAccessRoleArn=data_access_role_arn,
        InputDataConfig={
            "S3Uri": f"s3://{input_bucket}/{input_key}",
            "InputFormat": input_format.value,
        },
        OutputDataConfig={"S3Uri": f"s3://{output_bucket}/{output_key}"},
    )
    logger.info("Started topic modeling job %s.", response["JobId"])
except ClientError:
    logger.exception("Couldn't start topic modeling job.")
    raise
else:
    return response

```

- Einzelheiten zur API finden Sie [StartTopicsDetectionJob](#) in AWS SDK for Python (Boto3) API Reference.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Amazon Comprehend mit einem SDK verwenden AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

## Szenarien für Amazon Comprehend mit SDKs AWS

Die folgenden Codebeispiele zeigen Ihnen, wie Sie allgemeine Szenarien in Amazon Comprehend mit AWS SDKs implementieren. Diese Szenarien zeigen Ihnen, wie Sie bestimmte Aufgaben erledigen können, indem Sie mehrere Funktionen in Amazon Comprehend aufrufen. Jedes Szenario enthält einen Link zu GitHub, wo Sie Anweisungen zur Einrichtung und Ausführung des Codes finden.

### Beispiele

- [Erkennen Sie Dokumentelemente mit Amazon Comprehend und einem SDK AWS](#)
- [Führen Sie mithilfe eines SDK einen Amazon Comprehend-Themenmodellierungsjob für Beispieldaten aus AWS](#)
- [Trainieren Sie einen benutzerdefinierten Amazon Comprehend-Klassifikator und klassifizieren Sie Dokumente mithilfe eines SDK AWS](#)

## Erkennen Sie Dokumentelemente mit Amazon Comprehend und einem SDK AWS

Wie das aussehen kann, sehen Sie am nachfolgenden Beispielcode:

- Erkennt Sprachen, Entitäten und Schlüsselausdrücke in einem Dokument.
- Erkennt persönlich identifizierbare Informationen (PII) in einem Dokument.
- Ermitteln Sie die Stimmung in einem Dokument.
- Erkennt Syntaxelemente in einem Dokument.

## Python

### SDK für Python (Boto3)

#### Note

Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

Erstellen Sie eine Klasse, die Amazon Comprehend Comprehend-Aktionen umschließt.

```
import logging
from pprint import pprint
import boto3
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)

class ComprehendDetect:
    """Encapsulates Comprehend detection functions."""

    def __init__(self, comprehend_client):
        """
        :param comprehend_client: A Boto3 Comprehend client.
        """
        self.comprehend_client = comprehend_client

    def detect_languages(self, text):
        """
        Detects languages used in a document.

        :param text: The document to inspect.
        :return: The list of languages along with their confidence scores.
        """
        try:
            response = self.comprehend_client.detect_dominant_language(Text=text)
            languages = response["Languages"]
            logger.info("Detected %s languages.", len(languages))
        except ClientError:
            logger.exception("Couldn't detect languages.")
            raise
```



```
    else:
        return languages

def detect_entities(self, text, language_code):
    """
    Detects entities in a document. Entities can be things like people and
places
    or other common terms.

    :param text: The document to inspect.
    :param language_code: The language of the document.
    :return: The list of entities along with their confidence scores.
    """
    try:
        response = self.comprehend_client.detect_entities(
            Text=text, LanguageCode=language_code
        )
        entities = response["Entities"]
        logger.info("Detected %s entities.", len(entities))
    except ClientError:
        logger.exception("Couldn't detect entities.")
        raise
    else:
        return entities

def detect_key_phrases(self, text, language_code):
    """
    Detects key phrases in a document. A key phrase is typically a noun and
its
    modifiers.

    :param text: The document to inspect.
    :param language_code: The language of the document.
    :return: The list of key phrases along with their confidence scores.
    """
    try:
        response = self.comprehend_client.detect_key_phrases(
            Text=text, LanguageCode=language_code
        )
        phrases = response["KeyPhrases"]
        logger.info("Detected %s phrases.", len(phrases))
    except ClientError:
```

```
        logger.exception("Couldn't detect phrases.")
        raise
    else:
        return phrases

def detect_pii(self, text, language_code):
    """
    Detects personally identifiable information (PII) in a document. PII can
    be
    things like names, account numbers, or addresses.

    :param text: The document to inspect.
    :param language_code: The language of the document.
    :return: The list of PII entities along with their confidence scores.
    """
    try:
        response = self.comprehend_client.detect_pii_entities(
            Text=text, LanguageCode=language_code
        )
        entities = response["Entities"]
        logger.info("Detected %s PII entities.", len(entities))
    except ClientError:
        logger.exception("Couldn't detect PII entities.")
        raise
    else:
        return entities

def detect_sentiment(self, text, language_code):
    """
    Detects the overall sentiment expressed in a document. Sentiment can
    be positive, negative, neutral, or a mixture.

    :param text: The document to inspect.
    :param language_code: The language of the document.
    :return: The sentiments along with their confidence scores.
    """
    try:
        response = self.comprehend_client.detect_sentiment(
            Text=text, LanguageCode=language_code
        )
        logger.info("Detected primary sentiment %s.", response["Sentiment"])
    except ClientError:
```

```
        logger.exception("Couldn't detect sentiment.")
        raise
    else:
        return response

def detect_syntax(self, text, language_code):
    """
    Detects syntactical elements of a document. Syntax tokens are portions of
    text along with their use as parts of speech, such as nouns, verbs, and
    interjections.

    :param text: The document to inspect.
    :param language_code: The language of the document.
    :return: The list of syntax tokens along with their confidence scores.
    """
    try:
        response = self.comprehend_client.detect_syntax(
            Text=text, LanguageCode=language_code
        )
        tokens = response["SyntaxTokens"]
        logger.info("Detected %s syntax tokens.", len(tokens))
    except ClientError:
        logger.exception("Couldn't detect syntax.")
        raise
    else:
        return tokens
```

Rufen Sie Funktionen in der Wrapper-Klasse auf, um Entitäten, Phrasen und mehr in einem Dokument zu erkennen.

```
def usage_demo():
    print("-" * 88)
    print("Welcome to the Amazon Comprehend detection demo!")
    print("-" * 88)

    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")

    comp_detect = ComprehendDetect(boto3.client("comprehend"))
    with open("detect_sample.txt") as sample_file:
```

```
sample_text = sample_file.read()

demo_size = 3

print("Sample text used for this demo:")
print("-" * 88)
print(sample_text)
print("-" * 88)

print("Detecting languages.")
languages = comp_detect.detect_languages(sample_text)
pprint(languages)
lang_code = languages[0]["LanguageCode"]

print("Detecting entities.")
entities = comp_detect.detect_entities(sample_text, lang_code)
print(f"The first {demo_size} are:")
pprint(entities[:demo_size])

print("Detecting key phrases.")
phrases = comp_detect.detect_key_phrases(sample_text, lang_code)
print(f"The first {demo_size} are:")
pprint(phrases[:demo_size])

print("Detecting personally identifiable information (PII).")
pii_entities = comp_detect.detect_pii(sample_text, lang_code)
print(f"The first {demo_size} are:")
pprint(pii_entities[:demo_size])

print("Detecting sentiment.")
sentiment = comp_detect.detect_sentiment(sample_text, lang_code)
print(f"Sentiment: {sentiment['Sentiment']}")
print("SentimentScore:")
pprint(sentiment["SentimentScore"])

print("Detecting syntax elements.")
syntax_tokens = comp_detect.detect_syntax(sample_text, lang_code)
print(f"The first {demo_size} are:")
pprint(syntax_tokens[:demo_size])

print("Thanks for watching!")
print("-" * 88)
```

- Weitere API-Informationen finden Sie in den folgenden Themen der API-Referenz zum AWS -SDK für Python (Boto3).
  - [DetectDominantLanguage](#)
  - [DetectEntities](#)
  - [DetectKeyPhrases](#)
  - [DetectPiiEntities](#)
  - [DetectSentiment](#)
  - [DetectSyntax](#)

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Amazon Comprehend mit einem SDK verwenden AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

## Führen Sie mithilfe eines SDK einen Amazon Comprehend-Themenmodellierungsjob für Beispieldaten aus AWS

Wie das aussehen kann, sehen Sie am nachfolgenden Beispielcode:

- Führen Sie einen Amazon Comprehend Topic Modeling Job mit Beispieldaten aus.
- Informieren Sie sich über den Job.
- Extrahieren Sie Auftragsausgabedaten aus Amazon S3.

### Python

#### SDK für Python (Boto3)

##### Note

Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

Erstellen Sie eine Wrapper-Klasse, um Amazon Comprehend Topic Modeling Actions aufzurufen.

```
class ComprehendTopicModeler:
    """Encapsulates a Comprehend topic modeler."""

    def __init__(self, comprehend_client):
        """
        :param comprehend_client: A Boto3 Comprehend client.
        """
        self.comprehend_client = comprehend_client

    def start_job(
        self,
        job_name,
        input_bucket,
        input_key,
        input_format,
        output_bucket,
        output_key,
        data_access_role_arn,
    ):
        """
        Starts a topic modeling job. Input is read from the specified Amazon S3
        input bucket and written to the specified output bucket. Output data is
        stored
        in a tar archive compressed in gzip format. The job runs asynchronously,
        so you
        can call `describe_topics_detection_job` to get job status until it
        returns a status of SUCCEEDED.

        :param job_name: The name of the job.
        :param input_bucket: An Amazon S3 bucket that contains job input.
        :param input_key: The prefix used to find input data in the input
        bucket. If multiple objects have the same prefix,
        all
            of them are used.
        :param input_format: The format of the input data, either one document
        per
            file or one document per line.
        :param output_bucket: The Amazon S3 bucket where output data is written.
        :param output_key: The prefix prepended to the output data.
        :param data_access_role_arn: The Amazon Resource Name (ARN) of a role
        that
```

```
the grants Comprehend permission to read from
input bucket and write to the output bucket.
:return: Information about the job, including the job ID.
"""
try:
    response = self.comprehend_client.start_topics_detection_job(
        JobName=job_name,
        DataAccessRoleArn=data_access_role_arn,
        InputDataConfig={
            "S3Uri": f"s3://{input_bucket}/{input_key}",
            "InputFormat": input_format.value,
        },
        OutputDataConfig={"S3Uri": f"s3://{output_bucket}/{output_key}"},
    )
    logger.info("Started topic modeling job %s.", response["JobId"])
except ClientError:
    logger.exception("Couldn't start topic modeling job.")
    raise
else:
    return response

def describe_job(self, job_id):
    """
    Gets metadata about a topic modeling job.

    :param job_id: The ID of the job to look up.
    :return: Metadata about the job.
    """
    try:
        response = self.comprehend_client.describe_topics_detection_job(
            JobId=job_id
        )
        job = response["TopicsDetectionJobProperties"]
        logger.info("Got topic detection job %s.", job_id)
    except ClientError:
        logger.exception("Couldn't get topic detection job %s.", job_id)
        raise
    else:
        return job

def list_jobs(self):
```

```
"""
Lists topic modeling jobs for the current account.

:return: The list of jobs.
"""
try:
    response = self.comprehend_client.list_topics_detection_jobs()
    jobs = response["TopicsDetectionJobPropertiesList"]
    logger.info("Got %s topic detection jobs.", len(jobs))
except ClientError:
    logger.exception("Couldn't get topic detection jobs.")
    raise
else:
    return jobs
```

Verwenden Sie die Wrapper-Klasse, um einen Job zur Themenmodellierung auszuführen und Jobdaten abzurufen.

```
def usage_demo():
    print("-" * 88)
    print("Welcome to the Amazon Comprehend topic modeling demo!")
    print("-" * 88)

    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")

    input_prefix = "input/"
    output_prefix = "output/"
    demo_resources = ComprehendDemoResources(
        boto3.resource("s3"), boto3.resource("iam")
    )
    topic_modeler = ComprehendTopicModeler(boto3.client("comprehend"))

    print("Setting up storage and security resources needed for the demo.")
    demo_resources.setup("comprehend-topic-modeler-demo")
    print("Copying sample data from public bucket into input bucket.")
    demo_resources.bucket.copy(
        {"Bucket": "public-sample-us-west-2", "Key": "TopicModeling/Sample.txt"},
        f"{input_prefix}sample.txt",
    )
```



```
print("Starting topic modeling job on sample data.")
job_info = topic_modeler.start_job(
    "demo-topic-modeling-job",
    demo_resources.bucket.name,
    input_prefix,
    JobInputFormat.per_line,
    demo_resources.bucket.name,
    output_prefix,
    demo_resources.data_access_role.arn,
)

print(
    f"Waiting for job {job_info['JobId']} to complete. This typically takes "
    f"20 - 30 minutes."
)
job_waiter = JobCompleteWaiter(topic_modeler.comprehend_client)
job_waiter.wait(job_info["JobId"])

job = topic_modeler.describe_job(job_info["JobId"])
print(f"Job {job['JobId']} complete:")
pprint(job)

print(
    f"Getting job output data from the output Amazon S3 bucket: "
    f"{job['OutputDataConfig']['S3Uri']}."
)
job_output = demo_resources.extract_job_output(job)
lines = 10
print(f"First {lines} lines of document topics output:")
pprint(job_output["doc-topics.csv"]["data"][[:lines]])
print(f"First {lines} lines of terms output:")
pprint(job_output["topic-terms.csv"]["data"][[:lines]])

print("Cleaning up resources created for the demo.")
demo_resources.cleanup()

print("Thanks for watching!")
print("-" * 88)
```

- Weitere API-Informationen finden Sie in den folgenden Themen der API-Referenz zum AWS -SDK für Python (Boto3).
  - [DescribeTopicsDetectionJob](#)
  - [ListTopicsDetectionJobs](#)
  - [StartTopicsDetectionJob](#)

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Amazon Comprehend mit einem SDK verwenden AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

## Trainieren Sie einen benutzerdefinierten Amazon Comprehend-Klassifikator und klassifizieren Sie Dokumente mithilfe eines SDK AWS

Wie das aussehen kann, sehen Sie am nachfolgenden Beispielcode:

- Erstellen Sie einen Amazon Comprehend Multi-Label-Klassifikator.
- Trainieren Sie den Klassifikator anhand von Beispieldaten.
- Führen Sie einen Klassifizierungsjob für einen zweiten Datensatz aus.
- Extrahieren Sie die Auftragsausgabedaten aus Amazon S3.

### Python

#### SDK für Python (Boto3)

#### Note

Es gibt noch mehr dazu [GitHub](#). Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

Erstellen Sie eine Wrapper-Klasse, um Amazon Comprehend Document Classifier-Aktionen aufzurufen.

```
class ComprehendClassifier:
    """Encapsulates an Amazon Comprehend custom classifier."""

    def __init__(self, comprehend_client):
```

```
    """
    :param comprehend_client: A Boto3 Comprehend client.
    """
    self.comprehend_client = comprehend_client
    self.classifier_arn = None

def create(
    self,
    name,
    language_code,
    training_bucket,
    training_key,
    data_access_role_arn,
    mode,
):
    """
    Creates a custom classifier. After the classifier is created, it
    immediately
    starts training on the data found in the specified Amazon S3 bucket.
    Training
    can take 30 minutes or longer. The `describe_document_classifier`
    function
    can be used to get training status and returns a status of TRAINED when
    the
    classifier is ready to use.

    :param name: The name of the classifier.
    :param language_code: The language the classifier can operate on.
    :param training_bucket: The Amazon S3 bucket that contains the training
    data.
    :param training_key: The prefix used to find training data in the
    training
    bucket. If multiple objects have the same prefix,
    all
    of them are used.
    :param data_access_role_arn: The Amazon Resource Name (ARN) of a role
    that
    grants Comprehend permission to read from
    the
    training bucket.
    :return: The ARN of the newly created classifier.
    """
    try:
```

```
        response = self.comprehend_client.create_document_classifier(
            DocumentClassifierName=name,
            LanguageCode=language_code,
            InputDataConfig={"S3Uri": f"s3://{training_bucket}/
{training_key}"},
            DataAccessRoleArn=data_access_role_arn,
            Mode=mode.value,
        )
        self.classifier_arn = response["DocumentClassifierArn"]
        logger.info("Started classifier creation. Arn is: %s.",
self.classifier_arn)
    except ClientError:
        logger.exception("Couldn't create classifier %s.", name)
        raise
    else:
        return self.classifier_arn

def describe(self, classifier_arn=None):
    """
    Gets metadata about a custom classifier, including its current status.

    :param classifier_arn: The ARN of the classifier to look up.
    :return: Metadata about the classifier.
    """
    if classifier_arn is not None:
        self.classifier_arn = classifier_arn
    try:
        response = self.comprehend_client.describe_document_classifier(
            DocumentClassifierArn=self.classifier_arn
        )
        classifier = response["DocumentClassifierProperties"]
        logger.info("Got classifier %s.", self.classifier_arn)
    except ClientError:
        logger.exception("Couldn't get classifier %s.", self.classifier_arn)
        raise
    else:
        return classifier

def list(self):
    """
    Lists custom classifiers for the current account.
```

```
:return: The list of classifiers.
"""
try:
    response = self.comprehend_client.list_document_classifiers()
    classifiers = response["DocumentClassifierPropertiesList"]
    logger.info("Got %s classifiers.", len(classifiers))
except ClientError:
    logger.exception(
        "Couldn't get classifiers.",
    )
    raise
else:
    return classifiers

def delete(self):
    """
    Deletes the classifier.
    """
    try:
        self.comprehend_client.delete_document_classifier(
            DocumentClassifierArn=self.classifier_arn
        )
        logger.info("Deleted classifier %s.", self.classifier_arn)
        self.classifier_arn = None
    except ClientError:
        logger.exception("Couldn't deleted classifier %s.",
self.classifier_arn)
        raise

def start_job(
    self,
    job_name,
    input_bucket,
    input_key,
    input_format,
    output_bucket,
    output_key,
    data_access_role_arn,
):
    """
    Starts a classification job. The classifier must be trained or the job
    will fail. Input is read from the specified Amazon S3 input bucket and
```

```
written to the specified output bucket. Output data is stored in a tar
archive compressed in gzip format. The job runs asynchronously, so you
can
call `describe_document_classification_job` to get job status until it
returns a status of SUCCEEDED.

:param job_name: The name of the job.
:param input_bucket: The Amazon S3 bucket that contains input data.
:param input_key: The prefix used to find input data in the input
                  bucket. If multiple objects have the same prefix, all
                  of them are used.
:param input_format: The format of the input data, either one document
per
                    file or one document per line.
:param output_bucket: The Amazon S3 bucket where output data is written.
:param output_key: The prefix prepended to the output data.
:param data_access_role_arn: The Amazon Resource Name (ARN) of a role
that
                        grants Comprehend permission to read from
the
                        input bucket and write to the output bucket.
:return: Information about the job, including the job ID.
"""
try:
    response = self.comprehend_client.start_document_classification_job(
        DocumentClassifierArn=self.classifier_arn,
        JobName=job_name,
        InputDataConfig={
            "S3Uri": f"s3://{input_bucket}/{input_key}",
            "InputFormat": input_format.value,
        },
        OutputDataConfig={"S3Uri": f"s3://{output_bucket}/{output_key}"},
        DataAccessRoleArn=data_access_role_arn,
    )
    logger.info(
        "Document classification job %s is %s.", job_name,
response["JobStatus"]
    )
except ClientError:
    logger.exception("Couldn't start classification job %s.", job_name)
    raise
else:
    return response
```

```
def describe_job(self, job_id):
    """
    Gets metadata about a classification job.

    :param job_id: The ID of the job to look up.
    :return: Metadata about the job.
    """
    try:
        response =
self.comprehend_client.describe_document_classification_job(
            JobId=job_id
        )
        job = response["DocumentClassificationJobProperties"]
        logger.info("Got classification job %s.", job["JobName"])
    except ClientError:
        logger.exception("Couldn't get classification job %s.", job_id)
        raise
    else:
        return job

def list_jobs(self):
    """
    Lists the classification jobs for the current account.

    :return: The list of jobs.
    """
    try:
        response = self.comprehend_client.list_document_classification_jobs()
        jobs = response["DocumentClassificationJobPropertiesList"]
        logger.info("Got %s document classification jobs.", len(jobs))
    except ClientError:
        logger.exception(
            "Couldn't get document classification jobs.",
        )
        raise
    else:
        return jobs
```

Erstellen Sie eine Klasse, die Ihnen bei der Ausführung des Szenarios hilft.

```
class ClassifierDemo:
    """
    Encapsulates functions used to run the demonstration.
    """

    def __init__(self, demo_resources):
        """
        :param demo_resources: A ComprehendDemoResources class that manages
resources
                                for the demonstration.
        """
        self.demo_resources = demo_resources
        self.training_prefix = "training/"
        self.input_prefix = "input/"
        self.input_format = JobInputFormat.per_line
        self.output_prefix = "output/"

    def setup(self):
        """Creates AWS resources used by the demo."""
        self.demo_resources.setup("comprehend-classifier-demo")

    def cleanup(self):
        """Deletes AWS resources used by the demo."""
        self.demo_resources.cleanup()

    @staticmethod
    def _sanitize_text(text):
        """Removes characters that cause errors for the document parser."""
        return text.replace("\r", " ").replace("\n", " ").replace(",", ";")

    @staticmethod
    def _get_issues(query, issue_count):
        """
        Gets issues from GitHub using the specified query parameters.

        :param query: The query string used to request issues from the GitHub
API.
        :param issue_count: The number of issues to retrieve.
        :return: The list of issues retrieved from GitHub.
        """
        issues = []
        logger.info("Requesting issues from %s?%s.", GITHUB_SEARCH_URL, query)
```



```
response = requests.get(f"{GITHUB_SEARCH_URL}?
{query}&per_page={issue_count}")
if response.status_code == 200:
    issue_page = response.json()["items"]
    logger.info("Got %s issues.", len(issue_page))
    issues = [
        {
            "title": ClassifierDemo._sanitize_text(issue["title"]),
            "body": ClassifierDemo._sanitize_text(issue["body"]),
            "labels": {label["name"] for label in issue["labels"]},
        }
        for issue in issue_page
    ]
else:
    logger.error(
        "GitHub returned error code %s with message %s.",
        response.status_code,
        response.json(),
    )
logger.info("Found %s issues.", len(issues))
return issues

def get_training_issues(self, training_labels):
    """
    Gets issues used for training the custom classifier. Training issues are
    closed issues from the Boto3 repo that have known labels. Comprehend
    requires a minimum of ten training issues per label.

    :param training_labels: The issue labels to use for training.
    :return: The set of issues used for training.
    """
    issues = []
    per_label_count = 15
    for label in training_labels:
        issues += self._get_issues(
            f"q=type:issue+repo:boto/boto3+state:closed+label:{label}",
            per_label_count,
        )
        for issue in issues:
            issue["labels"] = issue["labels"].intersection(training_labels)
    return issues

def get_input_issues(self, training_labels):
    """
```

Gets input issues from GitHub. For demonstration purposes, input issues are open issues from the Boto3 repo with known labels, though in practice any issue could be submitted to the classifier for labeling.

:param training\_labels: The set of labels to query for.

:return: The set of issues used for input.

"""

```
issues = []
```

```
per_label_count = 5
```

```
for label in training_labels:
```

```
    issues += self._get_issues(
        f"q=type:issue+repo:boto/boto3+state:open+label:{label}",
        per_label_count,
    )
```

```
return issues
```

```
def upload_issue_data(self, issues, training=False):
```

"""

Uploads issue data to an Amazon S3 bucket, either for training or for input.

The data is first put into the format expected by Comprehend. For training,

the set of pipe-delimited labels is prepended to each document. For input, labels are not sent.

:param issues: The set of issues to upload to Amazon S3.

:param training: Indicates whether the issue data is used for training or input.

"""

```
try:
```

```
    obj_key = (
        self.training_prefix if training else self.input_prefix
    ) + "issues.txt"
```

```
    if training:
```

```
        issue_strings = [
            f"{'|'.join(issue['labels'])},{issue['title']}"
            f"{issue['body']}"
```

```
            for issue in issues
```

```
        ]
```

```
    else:
```

```
        issue_strings = [
            f"{issue['title']} {issue['body']}" for issue in issues
        ]
```

```
        issue_bytes = BytesIO("\n".join(issue_strings).encode("utf-8"))
```

```
        self.demo_resources.bucket.upload_fileobj(issue_bytes, obj_key)
        logger.info(
            "Uploaded data as %s to bucket %s.",
            obj_key,
            self.demo_resources.bucket.name,
        )
    except ClientError:
        logger.exception(
            "Couldn't upload data to bucket %s.",
            self.demo_resources.bucket.name
        )
        raise

    def extract_job_output(self, job):
        """Extracts job output from Amazon S3."""
        return self.demo_resources.extract_job_output(job)

    @staticmethod
    def reconcile_job_output(input_issues, output_dict):
        """
        Reconciles job output with the list of input issues. Because the input
        issues
        have known labels, these can be compared with the labels added by the
        classifier to judge the accuracy of the output.

        :param input_issues: The list of issues used as input.
        :param output_dict: The dictionary of data that is output by the
        classifier.
        :return: The list of reconciled input and output data.
        """
        reconciled = []
        for archive in output_dict.values():
            for line in archive["data"]:
                in_line = int(line["Line"])
                in_labels = input_issues[in_line]["labels"]
                out_labels = {
                    label["Name"]
                    for label in line["Labels"]
                    if float(label["Score"]) > 0.3
                }
                reconciled.append(
                    f"{line['File']}, line {in_line} has labels {in_labels}.\n"
                    f"\tClassifier assigned {out_labels}."
                )
        )
```

```
logger.info("Reconciled input and output labels.")
return reconciled
```

Trainieren Sie einen Klassifikator anhand einer Reihe von GitHub Problemen mit bekannten Bezeichnungen und senden Sie dann eine zweite Gruppe von GitHub Problemen an den Klassifikator, damit sie gekennzeichnet werden können.

```
def usage_demo():
    print("-" * 88)
    print("Welcome to the Amazon Comprehend custom document classifier demo!")
    print("-" * 88)

    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")

    comp_demo = ClassifierDemo(
        ComprehendDemoResources(boto3.resource("s3"), boto3.resource("iam"))
    )
    comp_classifier = ComprehendClassifier(boto3.client("comprehend"))
    classifier_trained_waiter = ClassifierTrainedWaiter(
        comp_classifier.comprehend_client
    )
    training_labels = {"bug", "feature-request", "dynamodb", "s3"}

    print("Setting up storage and security resources needed for the demo.")
    comp_demo.setup()

    print("Getting training data from GitHub and uploading it to Amazon S3.")
    training_issues = comp_demo.get_training_issues(training_labels)
    comp_demo.upload_issue_data(training_issues, True)

    classifier_name = "doc-example-classifier"
    print(f"Creating document classifier {classifier_name}.")
    comp_classifier.create(
        classifier_name,
        "en",
        comp_demo.demo_resources.bucket.name,
        comp_demo.training_prefix,
        comp_demo.demo_resources.data_access_role.arn,
        ClassifierMode.multi_label,
    )
```

```
print(
    f"Waiting until {classifier_name} is trained. This typically takes "
    f"30-40 minutes."
)
classifier_trained_waiter.wait(comp_classifier.classifier_arn)

print(f"Classifier {classifier_name} is trained:")
pprint(comp_classifier.describe())

print("Getting input data from GitHub and uploading it to Amazon S3.")
input_issues = comp_demo.get_input_issues(training_labels)
comp_demo.upload_issue_data(input_issues)

print("Starting classification job on input data.")
job_info = comp_classifier.start_job(
    "issue_classification_job",
    comp_demo.demo_resources.bucket.name,
    comp_demo.input_prefix,
    comp_demo.input_format,
    comp_demo.demo_resources.bucket.name,
    comp_demo.output_prefix,
    comp_demo.demo_resources.data_access_role.arn,
)
print(f"Waiting for job {job_info['JobId']} to complete.")
job_waiter = JobCompleteWaiter(comp_classifier.comprehend_client)
job_waiter.wait(job_info["JobId"])

job = comp_classifier.describe_job(job_info["JobId"])
print(f"Job {job['JobId']} complete:")
pprint(job)

print(
    f"Getting job output data from Amazon S3: "
    f"{job['OutputDataConfig']['S3Uri']}."
)
job_output = comp_demo.extract_job_output(job)
print("Job output:")
pprint(job_output)

print("Reconciling job output with labels from GitHub:")
reconciled_output = comp_demo.reconcile_job_output(input_issues, job_output)
print(*reconciled_output, sep="\n")
```

```
answer = input(f"Do you want to delete the classifier {classifier_name} (y/n)? ")
if answer.lower() == "y":
    print(f"Deleting {classifier_name}.")
    comp_classifier.delete()

print("Cleaning up resources created for the demo.")
comp_demo.cleanup()

print("Thanks for watching!")
print("-" * 88)
```

- Weitere API-Informationen finden Sie in den folgenden Themen der API-Referenz zum AWS-SDK für Python (Boto3).
  - [CreateDocumentClassifier](#)
  - [DeleteDocumentClassifier](#)
  - [DescribeDocumentClassificationJob](#)
  - [DescribeDocumentClassifier](#)
  - [ListDocumentClassificationJobs](#)
  - [ListDocumentClassifiers](#)
  - [StartDocumentClassificationJob](#)

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Amazon Comprehend mit einem SDK verwenden AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

## Serviceübergreifende Beispiele für Amazon Comprehend mit SDKs AWS

Die folgenden Beispielanwendungen verwenden AWS SDKs, um Amazon Comprehend mit anderen zu kombinieren. AWS-Services Jedes Beispiel enthält einen Link zu GitHub, wo Sie Anweisungen zum Einrichten und Ausführen der Anwendung finden.

### Beispiele

- [Erstellen einer Amazon-Transcribe-Streaming-App](#)
- [Erstellen Sie einen Amazon Lex Lex-Chatbot, um die Besucher Ihrer Website anzusprechen](#)
- [Erstellen Sie eine Webanwendung, die mithilfe von Amazon SQS Nachrichten sendet und abrufen](#)
- [Erstellen einer Anwendung, die Kundenfeedback analysiert und Audio generiert](#)
- [Erkennen Sie Entitäten in Text, der mithilfe eines AWS SDK aus einem Bild extrahiert wurde](#)

## Erstellen einer Amazon-Transcribe-Streaming-App

Das folgende Code-Beispiel zeigt, wie Sie eine App erstellen, die Live-Audio in Echtzeit aufzeichnet, transkribiert und übersetzt und die Ergebnisse per E-Mail sendet.

### JavaScript

#### SDK für JavaScript (v3)

Zeigt, wie Amazon Transcribe verwendet wird, um eine App zu erstellen, die Live-Audio in Echtzeit aufzeichnet, transkribiert und übersetzt und die Ergebnisse mit Amazon Simple Email Service (Amazon SES) per E-Mail sendet.

Den vollständigen Quellcode und Anweisungen zur Einrichtung und Ausführung finden Sie im vollständigen Beispiel unter [GitHub](#).

In diesem Beispiel verwendete Dienste

- Amazon Comprehend
- Amazon SES
- Amazon Transcribe
- Amazon Translate

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Amazon Comprehend mit einem SDK verwenden AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

## Erstellen Sie einen Amazon Lex Lex-Chatbot, um die Besucher Ihrer Website anzusprechen

Die folgenden Codebeispiele zeigen, wie Sie einen Chatbot erstellen, um die Besucher Ihrer Website anzusprechen.

## Java

### SDK für Java 2.x

Zeigt, wie Sie mithilfe der Amazon Lex Lex-API einen Chatbot innerhalb einer Webanwendung erstellen, um die Besucher Ihrer Website anzusprechen.

Den vollständigen Quellcode und Anweisungen zur Einrichtung und Ausführung finden Sie im vollständigen Beispiel unter [GitHub](#).

In diesem Beispiel verwendete Dienste

- Amazon Comprehend
- Amazon Lex
- Amazon Translate

## JavaScript

### SDK für JavaScript (v3)

Zeigt, wie Sie mithilfe der Amazon Lex Lex-API einen Chatbot innerhalb einer Webanwendung erstellen, um die Besucher Ihrer Website anzusprechen.

Den vollständigen Quellcode und Anweisungen zur Einrichtung und Ausführung finden Sie im vollständigen Beispiel „[Einen Amazon Lex-Chatbot erstellen](#)“ im AWS SDK for JavaScript Entwicklerhandbuch.

In diesem Beispiel verwendete Dienste

- Amazon Comprehend
- Amazon Lex
- Amazon Translate

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Amazon Comprehend mit einem SDK verwenden AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.



# Erstellen Sie eine Webanwendung, die mithilfe von Amazon SQS Nachrichten sendet und abrufen

Die folgenden Codebeispiele zeigen, wie Sie mithilfe von Amazon SQS eine Messaging-Anwendung erstellen.

## Java

### SDK für Java 2.x

Zeigt, wie die Amazon SQS SQS-API verwendet wird, um eine Spring-REST-API zu entwickeln, die Nachrichten sendet und abrufen.

Den vollständigen Quellcode und Anweisungen zur Einrichtung und Ausführung finden Sie im vollständigen Beispiel unter [GitHub](#)

In diesem Beispiel verwendete Dienste

- Amazon Comprehend
- Amazon SQS

## Kotlin

### SDK für Kotlin

Zeigt, wie die Amazon SQS SQS-API verwendet wird, um eine Spring-REST-API zu entwickeln, die Nachrichten sendet und abrufen.

Den vollständigen Quellcode und Anweisungen zur Einrichtung und Ausführung finden Sie im vollständigen Beispiel unter [GitHub](#)

In diesem Beispiel verwendete Dienste

- Amazon Comprehend
- Amazon SQS

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Amazon Comprehend mit einem SDK verwenden AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

# Erstellen einer Anwendung, die Kundenfeedback analysiert und Audio generiert

Die folgenden Codebeispiele zeigen, wie Sie eine Anwendung erstellen, die Kundenkommentarkarten analysiert, sie aus ihrer Originalsprache übersetzt, ihre Stimmung ermittelt und aus dem übersetzten Text eine Audiodatei generiert.

## .NET

### AWS SDK for .NET

Diese Beispielanwendung analysiert und speichert Kundenfeedback-Karten. Sie ist auf die Anforderungen eines fiktiven Hotels in New York City zugeschnitten. Das Hotel erhält Feedback von Gästen in Form von physischen Kommentarkarten in verschiedenen Sprachen. Dieses Feedback wird über einen Webclient in die App hochgeladen. Nachdem ein Bild einer Kommentarkarte hochgeladen wurde, werden folgende Schritte ausgeführt:

- Der Text wird mithilfe von Amazon Textract aus dem Bild extrahiert.
- Amazon Comprehend ermittelt die Stimmung und die Sprache des extrahierten Textes.
- Der extrahierte Text wird mithilfe von Amazon Translate ins Englische übersetzt.
- Amazon Polly generiert auf der Grundlage des extrahierten Texts eine Audiodatei.

Die vollständige App kann mithilfe des AWS CDK bereitgestellt werden. Den Quellcode und Anweisungen zur Bereitstellung finden Sie im Projekt unter [GitHub](#).

In diesem Beispiel verwendete Dienste

- Amazon Comprehend
- Lambda
- Amazon Polly
- Amazon Textract
- Amazon Translate

## Java

### SDK für Java 2.x

Diese Beispielanwendung analysiert und speichert Kundenfeedback-Karten. Sie ist auf die Anforderungen eines fiktiven Hotels in New York City zugeschnitten. Das Hotel erhält

Feedback von Gästen in Form von physischen Kommentarkarten in verschiedenen Sprachen. Dieses Feedback wird über einen Webclient in die App hochgeladen. Nachdem ein Bild einer Kommentarkarte hochgeladen wurde, werden folgende Schritte ausgeführt:

- Der Text wird mithilfe von Amazon Textract aus dem Bild extrahiert.
- Amazon Comprehend ermittelt die Stimmung und die Sprache des extrahierten Textes.
- Der extrahierte Text wird mithilfe von Amazon Translate ins Englische übersetzt.
- Amazon Polly generiert auf der Grundlage des extrahierten Texts eine Audiodatei.

Die vollständige App kann mithilfe des AWS CDK bereitgestellt werden. Den Quellcode und Anweisungen zur Bereitstellung finden Sie im Projekt unter [GitHub](#).

In diesem Beispiel verwendete Dienste

- Amazon Comprehend
- Lambda
- Amazon Polly
- Amazon Textract
- Amazon Translate

## JavaScript

### SDK für JavaScript (v3)

Diese Beispielanwendung analysiert und speichert Kundenfeedback-Karten. Sie ist auf die Anforderungen eines fiktiven Hotels in New York City zugeschnitten. Das Hotel erhält Feedback von Gästen in Form von physischen Kommentarkarten in verschiedenen Sprachen. Dieses Feedback wird über einen Webclient in die App hochgeladen. Nachdem ein Bild einer Kommentarkarte hochgeladen wurde, werden folgende Schritte ausgeführt:

- Der Text wird mithilfe von Amazon Textract aus dem Bild extrahiert.
- Amazon Comprehend ermittelt die Stimmung und die Sprache des extrahierten Textes.
- Der extrahierte Text wird mithilfe von Amazon Translate ins Englische übersetzt.
- Amazon Polly generiert auf der Grundlage des extrahierten Texts eine Audiodatei.

Die vollständige App kann mithilfe des AWS CDK bereitgestellt werden. Den Quellcode und Anweisungen zur Bereitstellung finden Sie im Projekt unter [GitHub](#). Die folgenden Auszüge zeigen, wie der innerhalb von Lambda-Funktionen verwendet AWS SDK for JavaScript wird.

```
import {
  ComprehendClient,
  DetectDominantLanguageCommand,
  DetectSentimentCommand,
} from "@aws-sdk/client-comprehend";

/**
 * Determine the language and sentiment of the extracted text.
 *
 * @param {{ source_text: string }} extractTextOutput
 */
export const handler = async (extractTextOutput) => {
  const comprehendClient = new ComprehendClient({});

  const detectDominantLanguageCommand = new DetectDominantLanguageCommand({
    Text: extractTextOutput.source_text,
  });

  // The source language is required for sentiment analysis and
  // translation in the next step.
  const { Languages } = await comprehendClient.send(
    detectDominantLanguageCommand,
  );

  const languageCode = Languages[0].LanguageCode;

  const detectSentimentCommand = new DetectSentimentCommand({
    Text: extractTextOutput.source_text,
    LanguageCode: languageCode,
  });

  const { Sentiment } = await comprehendClient.send(detectSentimentCommand);

  return {
    sentiment: Sentiment,
    language_code: languageCode,
  };
};
```

```
import {
  DetectDocumentTextCommand,
  TextractClient,
  Text,
} from "@aws-sdk/client-textract";
```

```
/**
 * Fetch the S3 object from the event and analyze it using Amazon Textract.
 *
 * @param {import("@types/aws-lambda").EventBridgeEvent<"Object Created">}
 eventBridgeS3Event
 */
export const handler = async (eventBridgeS3Event) => {
  const textractClient = new TextractClient();

  const detectDocumentTextCommand = new DetectDocumentTextCommand({
    Document: {
      S3Object: {
        Bucket: eventBridgeS3Event.bucket,
        Name: eventBridgeS3Event.object,
      },
    },
  });

  // Textract returns a list of blocks. A block can be a line, a page, word, etc.
  // Each block also contains geometry of the detected text.
  // For more information on the Block type, see https://docs.aws.amazon.com/textract/latest/dg/API\_Block.html.
  const { Blocks } = await textractClient.send(detectDocumentTextCommand);

  // For the purpose of this example, we are only interested in words.
  const extractedWords = Blocks.filter((b) => b.BlockType === "WORD").map(
    (b) => b.Text,
  );

  return extractedWords.join(" ");
};
```

```
import { PollyClient, SynthesizeSpeechCommand } from "@aws-sdk/client-polly";
import { S3Client } from "@aws-sdk/client-s3";
import { Upload } from "@aws-sdk/lib-storage";

/**
 * Synthesize an audio file from text.
 *
 * @param {{ bucket: string, translated_text: string, object: string }}
 sourceDestinationConfig
 */
```

```
export const handler = async (sourceDestinationConfig) => {
  const pollyClient = new PollyClient({});

  const synthesizeSpeechCommand = new SynthesizeSpeechCommand({
    Engine: "neural",
    Text: sourceDestinationConfig.translated_text,
    VoiceId: "Ruth",
    OutputFormat: "mp3",
  });

  const { AudioStream } = await pollyClient.send(synthesizeSpeechCommand);

  const audioKey = `${sourceDestinationConfig.object}.mp3`;

  // Store the audio file in S3.
  const s3Client = new S3Client();
  const upload = new Upload({
    client: s3Client,
    params: {
      Bucket: sourceDestinationConfig.bucket,
      Key: audioKey,
      Body: AudioStream,
      ContentType: "audio/mp3",
    },
  });

  await upload.done();
  return audioKey;
};
```

```
import {
  TranslateClient,
  TranslateTextCommand,
} from "@aws-sdk/client-translate";

/**
 * Translate the extracted text to English.
 *
 * @param {{ extracted_text: string, source_language_code: string }}
  textAndSourceLanguage
 */
export const handler = async (textAndSourceLanguage) => {
  const translateClient = new TranslateClient({});
```

```
const translateCommand = new TranslateTextCommand({
  SourceLanguageCode: textAndSourceLanguage.source_language_code,
  TargetLanguageCode: "en",
  Text: textAndSourceLanguage.extracted_text,
});

const { TranslatedText } = await translateClient.send(translateCommand);

return { translated_text: TranslatedText };
};
```

In diesem Beispiel verwendete Dienste

- Amazon Comprehend
- Lambda
- Amazon Polly
- Amazon Textract
- Amazon Translate

## Ruby

### SDK für Ruby

Diese Beispielanwendung analysiert und speichert Kundenfeedback-Karten. Sie ist auf die Anforderungen eines fiktiven Hotels in New York City zugeschnitten. Das Hotel erhält Feedback von Gästen in Form von physischen Kommentarkarten in verschiedenen Sprachen. Dieses Feedback wird über einen Webclient in die App hochgeladen. Nachdem ein Bild einer Kommentarkarte hochgeladen wurde, werden folgende Schritte ausgeführt:

- Der Text wird mithilfe von Amazon Textract aus dem Bild extrahiert.
- Amazon Comprehend ermittelt die Stimmung und die Sprache des extrahierten Textes.
- Der extrahierte Text wird mithilfe von Amazon Translate ins Englische übersetzt.
- Amazon Polly generiert auf der Grundlage des extrahierten Textes eine Audiodatei.

Die vollständige App kann mithilfe des AWS CDK bereitgestellt werden. Quellcode und Anweisungen zur Bereitstellung finden Sie im Projekt unter. [GitHub](#)

In diesem Beispiel verwendete Dienste

- Amazon Comprehend

- Lambda
- Amazon Polly
- Amazon Textract
- Amazon Translate

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Amazon Comprehend mit einem SDK verwenden AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

## Erkennen Sie Entitäten in Text, der mithilfe eines AWS SDK aus einem Bild extrahiert wurde

Das folgende Codebeispiel zeigt, wie Amazon Comprehend verwendet wird, um Entitäten in Text zu erkennen, der von Amazon Textract aus einem in Amazon S3 gespeicherten Bild extrahiert wurde.

### Python

#### SDK für Python (Boto3)

Zeigt, wie das AWS SDK for Python (Boto3) in einem Jupyter-Notizbuch verwendet wird, um Entitäten in Text zu erkennen, der aus einem Bild extrahiert wurde. In diesem Beispiel extrahiert Amazon Textract Text aus einem Bild, das in Amazon Simple Storage Service (Amazon S3) und Amazon Comprehend gespeichert ist, um Entitäten im extrahierten Text zu erkennen.

Dieses Beispiel ist ein Jupyter Notebook und muss in einer Umgebung ausgeführt werden, die Notebooks hosten kann. Anweisungen zur Ausführung des Beispiels mit Amazon SageMaker finden Sie in den Anweisungen in der Datei [TextractAndComprehendNotebook.ipynb](#).

Den vollständigen Quellcode und Anweisungen zur Einrichtung und Ausführung finden Sie im vollständigen Beispiel unter [GitHub](#).

In diesem Beispiel verwendete Dienste

- Amazon Comprehend
- Amazon S3
- Amazon Textract



Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Amazon Comprehend mit einem SDK verwenden AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

# Sicherheit in Amazon Comprehend

Die Sicherheit in der Cloud hat für AWS höchste Priorität. Als AWS-Kunde profitieren Sie von einer Rechenzentrums- und Netzwerkarchitektur, die zur Erfüllung der Anforderungen von Organisationen entwickelt wurden, für die Sicherheit eine kritische Bedeutung hat.

Sicherheit gilt zwischen AWS und Ihnen eine geteilte Verantwortung. Das [Modell der geteilten Verantwortung](#) dies als Sicherheit der Cloud und Sicherheit in der Cloud:

- Sicherheit der Cloud selbst – AWS ist dafür verantwortlich, die Infrastruktur zu schützen, mit der AWS-Services in der AWS Cloud ausgeführt werden. AWS stellt Ihnen außerdem Services bereit, die Sie sicher nutzen können. Auditoren von Drittanbietern testen und überprüfen die Effektivität unserer Sicherheitsmaßnahmen im Rahmen der [AWS-Compliance-Programme](#) regelmäßig. Informationen zu den Compliance-Programmen, die für Amazon Comprehend gelten, finden Sie unter [AWS Im Rahmen des Compliance-Programms zugelassene -ServicesIm](#).
- Sicherheit in der Cloud – Ihr Verantwortungsumfang wird durch den AWS-Dienst bestimmt, den Sie verwenden. Sie sind auch für andere Faktoren verantwortlich, etwa für die Vertraulichkeit Ihrer Daten, für die Anforderungen Ihres Unternehmens und für die geltenden Gesetze und Vorschriften.

Diese Dokumentation hilft Ihnen zu verstehen, wie Sie das Modell der geteilten Verantwortung bei der Verwendung von Amazon Comprehend einsetzen können. Die folgenden Themen zeigen Ihnen, wie Sie Amazon Comprehend zur Erfüllung Ihrer Sicherheits- und Compliance-Ziele konfigurieren. Sie erfahren auch, wie Sie andere -AWSServices verwenden, die Sie bei der Überwachung und Sicherung Ihrer Amazon Comprehend-Ressourcen unterstützen.

## Themen

- [Datenschutz in Amazon Comprehend](#)
- [Identity and Access Management für Amazon Comprehend](#)
- [Protokollieren von Amazon Comprehend-API-Aufrufen mit AWS CloudTrail](#)
- [Compliance-Validierung für Amazon Comprehend](#)
- [Ausfallsicherheit in Amazon Comprehend](#)
- [Infrastruktursicherheit in Amazon Comprehend](#)

# Datenschutz in Amazon Comprehend

Das Modell der AWS geteilten gilt für den Datenschutz in Amazon Comprehend . <https://aws.amazon.com/compliance/shared-responsibility-model/> Wie in diesem Modell beschrieben, ist AWS für den Schutz der globalen Infrastruktur verantwortlich, in der die gesamte AWS Cloud ausgeführt wird. Sie sind dafür verantwortlich, die Kontrolle über Ihre in dieser Infrastruktur gehosteten Inhalte zu behalten. Sie sind auch für die Sicherheitskonfiguration und die Verwaltungsaufgaben für die von Ihnen verwendeten AWS-Services verantwortlich. Weitere Informationen zum Datenschutz finden Sie unter [Häufig gestellte Fragen zum Datenschutz](#). Informationen zum Datenschutz in Europa finden Sie im Blog-Beitrag [AWS-Modell der geteilten Verantwortung und in der DSGVO](#) im AWS-Sicherheitsblog.

Aus Datenschutzgründen empfehlen wir, AWS-Konto-Anmeldeinformationen zu schützen und einzelne Benutzer mit AWS IAM Identity Center oder AWS Identity and Access Management (IAM) einzurichten. So erhält jeder Benutzer nur die Berechtigungen, die zum Durchführen seiner Aufgaben erforderlich sind. Außerdem empfehlen wir, die Daten mit folgenden Methoden zu schützen:

- Verwenden Sie für jedes Konto die Multi-Faktor Authentifizierung (MFA).
- Verwenden Sie SSL/TLS für die Kommunikation mit AWS-Ressourcen. Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Richten Sie die API und die Protokollierung von Benutzeraktivitäten mit AWS CloudTrail ein.
- Verwenden Sie AWS-Verschlüsselungslösungen zusammen mit allen Standardsicherheitskontrollen in AWS-Services.
- Verwenden Sie erweiterte verwaltete Sicherheitsservices wie Amazon Macie, die dabei helfen, in Amazon S3 gespeicherte persönliche Daten zu erkennen und zu schützen.
- Wenn Sie für den Zugriff auf AWS über eine Befehlszeilenschnittstelle oder über eine API FIPS 140-2-validierte kryptografische Module benötigen, verwenden Sie einen FIPS-Endpunkt. Weitere Informationen über verfügbare FIPS-Endpunkte finden Sie unter [Federal Information Processing Standard \(FIPS\) 140-2](#).

Wir empfehlen dringend, in Freitextfeldern, z. B. im Feld Name, keine vertraulichen oder sensiblen Informationen wie die E-Mail-Adressen Ihrer Kunden einzugeben. Dies gilt auch, wenn Sie mit Amazon Comprehend oder anderen AWS-Services über die Konsole, API/AWS CLI, oder AWS SDKs arbeiten. Alle Daten, die Sie in Tags oder Freitextfelder eingeben, die für Namen verwendet werden, können für Abrechnungs- oder Diagnoseprotokolle verwendet werden. Wenn Sie eine URL für einen

externen Server bereitstellen, empfehlen wir dringend, keine Anmeldeinformationen zur Validierung Ihrer Anforderung an den betreffenden Server in die URL einzuschließen.

## Themen

- [KMS-Verschlüsselung in Amazon Comprehend](#)
- [Serviceübergreifende Confused-Deputy-Prävention](#)
- [Schützen Sie Jobs mithilfe einer Amazon Virtual Private Cloud](#)
- [Amazon Comprehend und Schnittstellen-VPC-Endpunkte \(AWS PrivateLink\)](#)

## KMS-Verschlüsselung in Amazon Comprehend

Amazon Comprehend arbeitet mit AWS Key Management Service (AWS KMS), um eine verbesserte Verschlüsselung für Ihre Daten bereitzustellen. Mit Amazon S3 können Sie Ihre Eingabedokumente bereits verschlüsseln, wenn Sie eine Textanalyse, Themenmodellierung oder einen benutzerdefinierten Amazon Comprehend-Auftrag erstellen. Durch die Integration mit AWS KMS können Sie die Daten im Speichervolumen für Start\*- und Create\*-Aufträge verschlüsseln und die Ausgabeergebnisse von Start\*-Aufträgen mit Ihrem eigenen KMS-Schlüssel verschlüsseln.

Für die verschlüsselt AWS Management Console Amazon Comprehend benutzerdefinierte Modelle mit einem eigenen KMS-Schlüssel. Für die kann AWS CLI Amazon Comprehend benutzerdefinierte Modelle entweder mit einem eigenen KMS-Schlüssel oder einem bereitgestellten kundenverwalteten Schlüssel (CMK) verschlüsseln.

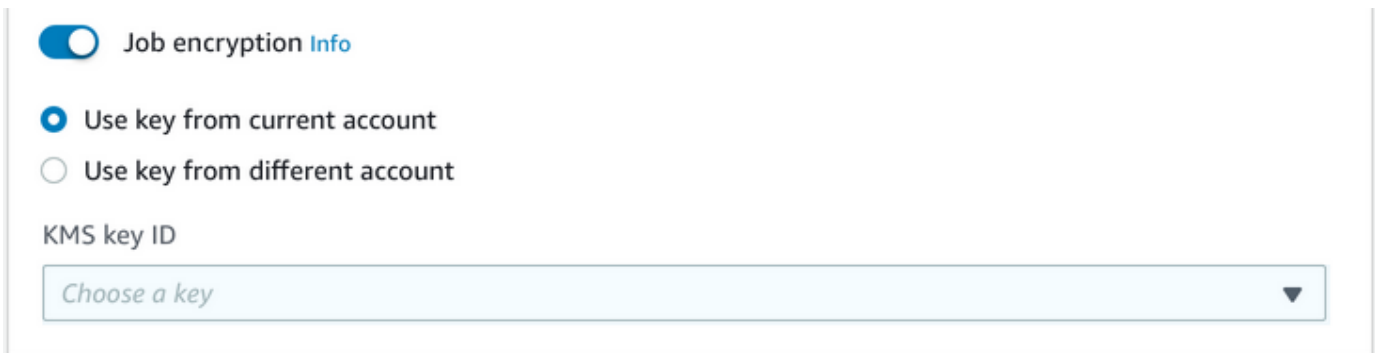
### KMS-Verschlüsselung mit der AWS Management Console

Bei Verwendung der Konsole sind zwei Verschlüsselungsoptionen verfügbar:

- Volume-Verschlüsselung
- Ausgabeergebnisverschlüsselung

So aktivieren Sie die Volume-Verschlüsselung

1. Wählen Sie unter Auftragseinstellungen die Option Auftragsverschlüsselung aus.



Job encryption [Info](#)

Use key from current account

Use key from different account

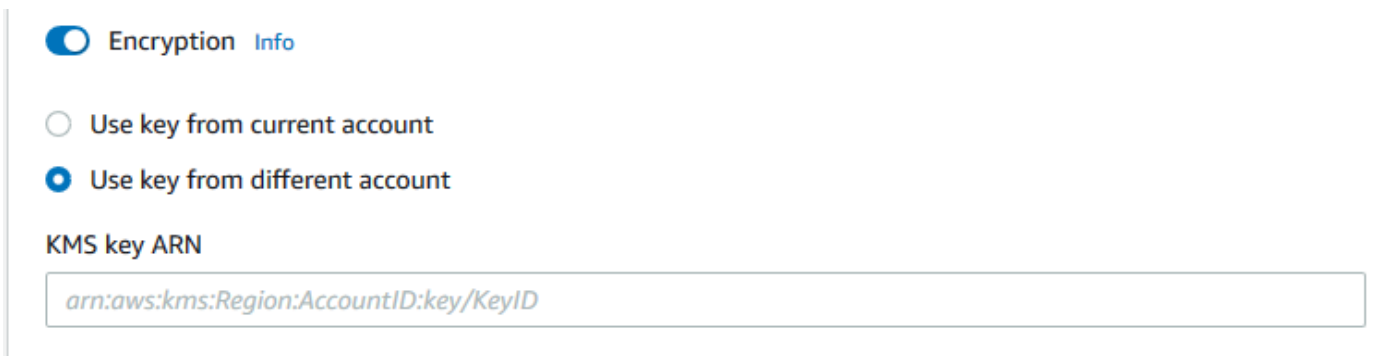
KMS key ID

Choose a key ▼

2. Wählen Sie aus, ob der kundenverwaltete Schlüssel (CMK) von KMS aus dem Konto stammt, das Sie gerade verwenden, oder aus einem anderen Konto. Wenn Sie einen Schlüssel aus dem aktuellen Konto verwenden möchten, wählen Sie den Schlüsselalias aus der KMS-Schlüssel-ID aus. Wenn Sie einen Schlüssel aus einem anderen Konto verwenden, müssen Sie den ARN des Schlüssels eingeben.

So aktivieren Sie die Verschlüsselung der Ausgabeergebnisse

1. Wählen Sie unter Ausgabeeinstellungen die Option Verschlüsselung aus.



Encryption [Info](#)

Use key from current account

Use key from different account

KMS key ARN

arn:aws:kms:Region:AccountID:key/KeyID

2. Wählen Sie aus, ob der vom Kunden verwaltete Schlüssel (CMK) von dem Konto stammt, das Sie derzeit verwenden, oder von einem anderen Konto. Wenn Sie einen Schlüssel aus dem aktuellen Konto verwenden möchten, wählen Sie die Schlüssel-ID aus der KMS-Schlüssel-ID aus. Wenn Sie einen Schlüssel aus einem anderen Konto verwenden, müssen Sie den ARN des Schlüssels eingeben.

Wenn Sie zuvor die Verschlüsselung mit SSE-KMS auf Ihren S3-Eingabedokumenten eingerichtet haben, kann Ihnen dies zusätzliche Sicherheit bieten. In diesem Fall muss die verwendete IAM-Rolle jedoch über die `kms:Decrypt` Berechtigung für den KMS-Schlüssel verfügen, mit dem die Eingabedokumente verschlüsselt sind. Weitere Informationen finden Sie unter [Erforderliche Berechtigungen für die Verwendung der KMS-Verschlüsselung](#).

## KMS-Verschlüsselung mit -API-Operationen

Alle Amazon Comprehend- `Start*` und `Create*`-API-Operationen unterstützen KMS-verschlüsselte Eingabedokumente. `Describe*` und `List*`-API-Operationen geben den `KmsKeyId` in `zurückOutputDataConfig`, wenn der ursprüngliche Auftrag als Eingabe `KmsKeyId` bereitgestellt wurde. Wenn es nicht als Eingabe bereitgestellt wurde, wird es nicht zurückgegeben.

Dies ist im folgenden AWS CLI-Beispiel mit der [-StartEntitiesDetectionJob](#) Operation zu sehen:

```
aws comprehend start-entities-detection-job \  
  --region region \  
  --data-access-role-arn "data access role arn" \  
  --entity-recognizer-arn "entity recognizer arn" \  
  --input-data-config "S3Uri=s3://Bucket Name/Bucket Path" \  
  --job-name job name \  
  --language-code en \  
  --output-data-config "KmsKeyId=Output S3 KMS key ID" "S3Uri=s3://Bucket  
Name/Bucket Path/" \  
  --volumekmskeyid "Volume KMS key ID"
```

### Note

Dieses Beispiel ist für Unix, Linux und macOS formatiert. Ersetzen Sie unter Windows den umgekehrten Schrägstrich (\), das Unix-Fortsetzungszeichen, am Ende jeder Zeile durch ein Caret-Zeichen oder Zirkumflex (^).

## Kundenverwaltete Schlüssel (CMK)-Verschlüsselung mit -API-Operationen

Die benutzerdefinierten Modell-API-Operationen von Amazon Comprehend, `CreateDocumentClassifier`, und `CreateEndpoint`, unterstützen die `CreateEntityRecognizer` Verschlüsselung mit vom Kunden verwalteten Schlüsseln über die AWS CLI.

Sie benötigen eine IAM-Richtlinie, damit ein Prinzipal vom Kunden verwaltete Schlüssel verwenden oder verwalten kann. Diese Schlüssel werden im `-ResourceElement` der Richtlinienanweisung angegeben. Beschränken Sie als bewährte Methode vom Kunden verwaltete Schlüssel auf diejenigen, die die Prinzipale in Ihrer Richtlinienanweisung verwenden müssen.

Im folgenden AWS CLI-Beispiel wird ein benutzerdefinierter Entity Recognizer mit Modellverschlüsselung unter Verwendung der [CreateEntityRecognizer](#) Operation erstellt:

```
aws comprehend create-entity-recognizer \  
  --recognizer-name name \  
  --data-access-role-arn data access role arn \  
  --language-code en \  
  --model-kms-key-id Model KMS Key ID \  
  --input-data-config file:///path/input-data-config.json
```

### Note

Dieses Beispiel ist für Unix, Linux und macOS formatiert. Ersetzen Sie unter Windows den umgekehrten Schrägstrich (\), das Unix-Fortsetzungszeichen, am Ende jeder Zeile durch ein Caret-Zeichen oder Zirkumflex (^).

## Serviceübergreifende Confused-Deputy-Prävention

Das Confused-Deputy-Problem ist ein Sicherheitsproblem, bei dem eine juristische Stelle, die nicht über die Berechtigung zum Ausführen einer Aktion verfügt, eine privilegiertere juristische Stelle zwingen kann, die Aktion auszuführen. In AWS kann der serviceübergreifende Identitätswechsel zu Confused-Deputy-Problem führen. Ein dienstübergreifender Identitätswechsel kann auftreten, wenn ein Dienst (der Anruf-Dienst) einen anderen Dienst anruft (den aufgerufenen Dienst). Der aufrufende Service kann manipuliert werden, um seine Berechtigungen zu verwenden, um Aktionen auf die Ressourcen eines anderen Kunden auszuführen, für die er sonst keine Zugriffsberechtigung haben sollte. Um dies zu verhindern, bietet AWS Tools, mit denen Sie Ihre Daten für alle Services mit Serviceprinzipalen schützen können, die Zugriff auf Ressourcen in Ihrem Konto erhalten haben.

Wir empfehlen die Verwendung der [aws:SourceAccount](#) globalen Bedingungskontextschlüssel [aws:SourceArn](#) und in Ressourcenrichtlinien, um die Berechtigungen einzuschränken, die Amazon Comprehend einem anderen Service für die Ressource erteilt. Wenn Sie beide globalen Bedingungskontextschlüssel verwenden, müssen der `aws:SourceAccount`-Wert und das Konto im `aws:SourceArn`-Wert dieselbe Konto-ID verwenden, wenn sie in derselben Richtlinienanweisung verwendet werden.

Der effektivste Weg, um sich vor dem Confused-Deputy-Problem zu schützen, ist die Verwendung des globalen Bedingungskontext-Schlüssels `aws:SourceArn` mit dem vollständigen

ARN der Ressource. Wenn Sie den vollständigen ARN der Ressource nicht kennen oder wenn Sie mehrere Ressourcen angeben, verwenden Sie den globalen Bedingungskontext-Schlüssel `aws:SourceArn` mit Platzhaltern (\*) für die unbekanntenen Teile des ARN. Beispiel:

```
arn:aws:servicename:123456789012:*
```

## Verwenden des Quellkontos

Das folgende Beispiel zeigt, wie Sie den `aws:SourceAccount` globalen Bedingungskontextschlüssel in Amazon Comprehend verwenden können.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Sid": "ConfusedDeputyPreventionExamplePolicy",
    "Effect": "Allow",
    "Principal": {
      "Service": "comprehend.amazonaws.com"
    },
    "Action": "sts:AssumeRole",
    "Condition": {
      "StringEquals": {
        "aws:SourceAccount": "111122223333"
      }
    }
  }
}
```

## Vertrauensrichtlinie für Endpunkte verschlüsselter Modelle

Sie müssen eine Vertrauensrichtlinie erstellen, um einen Endpunkt für ein verschlüsseltes Modell zu erstellen oder zu aktualisieren. Legen Sie den Wert `aws:SourceAccount` auf Ihre Konto-ID fest. Wenn Sie die `ArnEquals` Bedingung verwenden, legen Sie den `aws:SourceArn` Wert auf den ARN des Endpunkts fest.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "comprehend.amazonaws.com"
      }
    }
  ]
}
```



```

    },
    "Action": "sts:AssumeRole",
    "Condition": {
      "StringEquals": {
        "aws:SourceAccount": "111122223333"
      },
      "ArnEquals": {
        "aws:SourceArn": "arn:aws:comprehend:us-west-2:111122223333:document-
classifier-endpoint/endpoint-name"
      }
    }
  }
]
}

```

## Erstellen eines benutzerdefinierten Modells

Sie müssen eine Vertrauensrichtlinie erstellen, um ein benutzerdefiniertes Modell zu erstellen. Legen Sie den Wert `aws:SourceAccount` auf Ihre Konto-ID fest. Wenn Sie die `ArnEquals` Bedingung verwenden, legen Sie den `aws:SourceArn` Wert auf den ARN der benutzerdefinierten Modellversion fest.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "comprehend.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "111122223333"
        },
        "ArnEquals": {
          "aws:SourceArn": "arn:aws:comprehend:us-west-2:111122223333:
document-classifier/smallest-classifier-test/
version/version-name"
        }
      }
    }
  ]
}

```

```
]
}
```

## Schützen Sie Jobs mithilfe einer Amazon Virtual Private Cloud

Amazon Comprehend verwendet eine Vielzahl von Sicherheitsmaßnahmen, um die Sicherheit Ihrer Daten in unseren Jobcontainern zu gewährleisten, in denen sie gespeichert werden, während sie von Amazon Comprehend verwendet werden. Job-Container greifen jedoch über das Internet auf AWS Ressourcen zu — wie die Amazon S3 S3-Buckets, in denen Sie Daten und Modellartefakte speichern.

Um den Zugriff auf Ihre Daten zu kontrollieren, empfehlen wir Ihnen, eine Virtual Private Cloud (VPC) zu erstellen und diese so zu konfigurieren, dass die Daten und Container nicht über das Internet zugänglich sind. Informationen zum Erstellen und Konfigurieren einer VPC finden Sie unter [Erste Schritte mit Amazon VPC](#) im Amazon VPC Benutzerhandbuch. Die Verwendung einer VPC trägt zum Schutz Ihrer Daten bei, da Sie Ihre VPC so konfigurieren können, dass sie nicht mit dem Internet verbunden ist. Die Verwendung einer VPC ermöglicht es Ihnen auch, den gesamten Netzwerkverkehr in und aus unseren Jobcontainern mithilfe von VPC-Flow-Logs zu überwachen. Weitere Informationen finden Sie unter [VPC-Flow-Protokolle](#) im Amazon-VPC-Benutzerhandbuch.

Sie geben Ihre VPC-Konfiguration an, wenn Sie einen Job erstellen, indem Sie die Subnetze und Sicherheitsgruppen angeben. Wenn Sie die Subnetze und Sicherheitsgruppen angeben, erstellt Amazon Comprehend Elastic Network Interfaces (ENIs), die Ihren Sicherheitsgruppen in einem der Subnetze zugeordnet sind. ENIs ermöglichen es unseren Jobcontainern, sich mit Ressourcen in Ihrer VPC zu verbinden. Weitere Informationen über ENIs finden Sie unter [Elastic-Network-Schnittstellen](#) im Amazon-VPC-Benutzerhandbuch.

### Note

Für Jobs können Sie nur Subnetze mit einer Standard-Tenancy-VPC konfigurieren, in der Ihre Instance auf gemeinsam genutzter Hardware läuft. Weitere Informationen zum Tenancy-Attribut für VPCs finden Sie unter [Dedicated Instances](#) im Amazon EC2 EC2-Benutzerhandbuch.

## Einen Job für den Amazon VPC-Zugriff konfigurieren

Um Subnetze und Sicherheitsgruppen in Ihrer VPC anzugeben, verwenden Sie den `VpcConfig` Anforderungsparameter der entsprechenden API oder geben Sie diese Informationen an, wenn

Sie einen Job in der Amazon Comprehend Comprehend-Konsole erstellen. Amazon Comprehend verwendet diese Informationen, um ENIs zu erstellen und sie an unsere Jobcontainer anzuhängen. Die ENIs stellen unseren Jobcontainern eine Netzwerkverbindung innerhalb Ihrer VPC zur Verfügung, die nicht mit dem Internet verbunden ist.

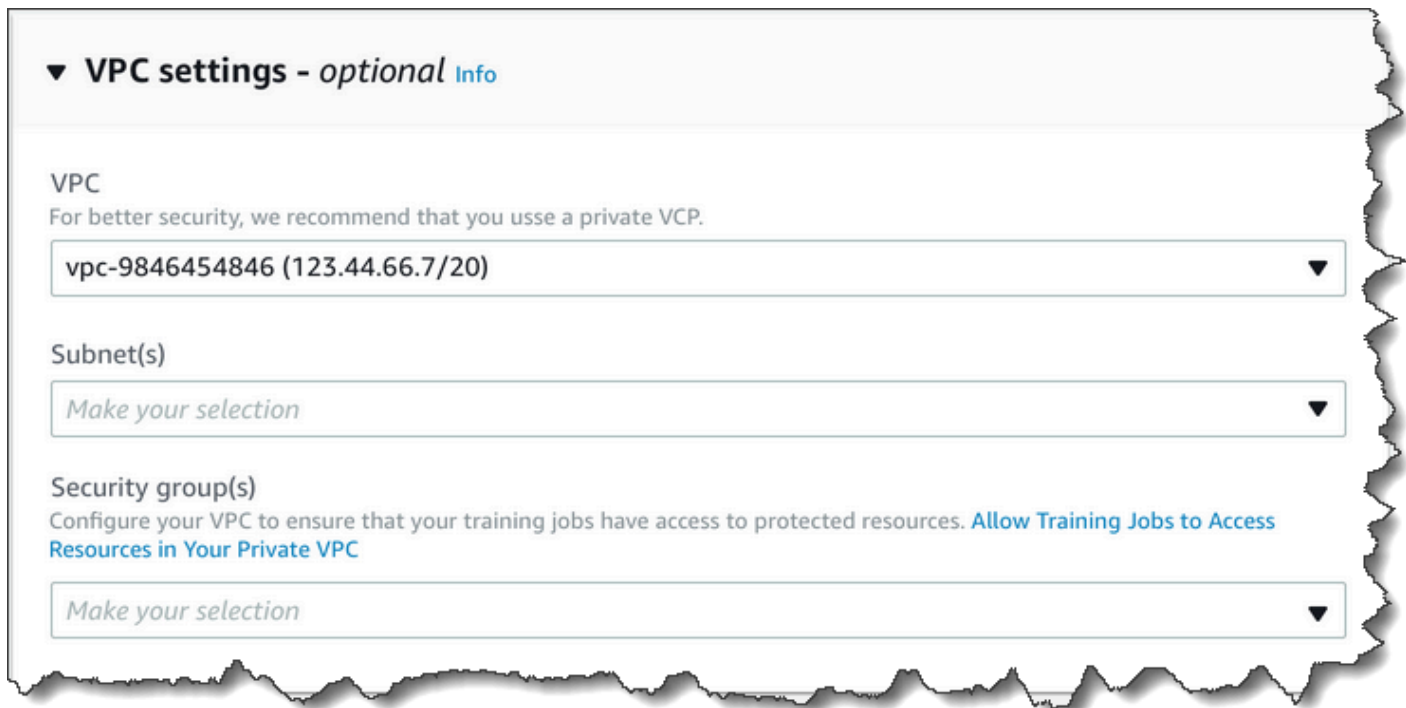
Die folgenden APIs enthalten den VpcConfig Anforderungsparameter:

- Create\*APIs: [CreateDocumentClassifier](#), [CreateEntityRecognizer](#)
- Start\*APIs: [StartDocumentClassificationJob](#), [StartDominantLanguageDetectionJob](#), [StartEntitiesDetectionJob](#), [StartKeyPhrasesDetectionJob](#), [StartSentimentDetectionJob](#), [StartTargetedSentimentDetectionJob](#), [StartTopicsDetectionJob](#)

Im Folgenden finden Sie ein Beispiel für den VpcConfig Parameter, den Sie in Ihren API-Aufruf aufnehmen:

```
"VpcConfig": {
  "SecurityGroupIds": [
    " sg-0123456789abcdef0"
  ],
  "Subnets": [
    "subnet-0123456789abcdef0",
    "subnet-0123456789abcdef1",
    "subnet-0123456789abcdef2"
  ]
}
```

Um eine VPC von der Amazon Comprehend Comprehend-Konsole aus zu konfigurieren, wählen Sie bei der Erstellung des Jobs die Konfigurationsdetails aus dem optionalen Abschnitt VPC-Einstellungen aus.



▼ **VPC settings - optional** [Info](#)

**VPC**  
For better security, we recommend that you use a private VPC.

vpc-9846454846 (123.44.66.7/20) ▼

**Subnet(s)**

Make your selection ▼

**Security group(s)**  
Configure your VPC to ensure that your training jobs have access to protected resources. [Allow Training Jobs to Access Resources in Your Private VPC](#)

Make your selection ▼

## Konfigurieren Sie Ihre VPC für Amazon Comprehend Comprehend-Jobs

Beachten Sie bei der Konfiguration der VPC für Ihre Amazon Comprehend Comprehend-Jobs die folgenden Richtlinien. Informationen zum Einrichten einer VPC finden Sie unter [Arbeiten mit VPCs und Subnetzen](#) im Amazon VPC Benutzerhandbuch.

Stellen Sie sicher, dass die Subnetze über genügend IP-Adressen verfügen

Ihre VPC-Subnetze sollten mindestens zwei private IP-Adressen für jede Instance in einem Job haben. Weitere Informationen finden Sie unter [Dimensionierung der VPC und der Subnetze für IPv4](#) im Amazon VPC Benutzerhandbuch.

### Erstellen Sie einen Amazon S3 S3-VPC-Endpunkt

Wenn Sie Ihre VPC so konfigurieren, dass Job-Container keinen Zugriff auf das Internet haben, können sie keine Verbindung zu den Amazon S3 S3-Buckets herstellen, die Ihre Daten enthalten, es sei denn, Sie erstellen einen VPC-Endpunkt, der den Zugriff ermöglicht. Durch die Erstellung eines VPC-Endpunkts ermöglichen Sie den Job-Containern den Zugriff auf Ihre Daten während Trainings- und Analysejobs.

Wenn Sie den VPC-Endpunkt erstellen, konfigurieren Sie diese Werte:

- Wählen Sie als AWS Dienstkategorie Dienste aus

- Geben Sie den Dienst als `com.amazonaws.region.s3` an.
- Wählen Sie Gateway als VPC-Endpunkttyp aus.

Wenn Sie AWS CloudFormation zum Erstellen des VPC-Endpunkts verwenden, folgen Sie der [AWS CloudFormation VPCEndpoint-Dokumentation](#). Das folgende Beispiel zeigt die `vpcEndpoint`-Konfiguration in einer Vorlage. AWS CloudFormation

```
VpcEndpoint:
  Type: AWS::EC2::VPCEndpoint
  Properties:
    PolicyDocument:
      Version: '2012-10-17'
      Statement:
        - Action:
            - s3:GetObject
            - s3:PutObject
            - s3:ListBucket
            - s3:GetBucketLocation
            - s3:DeleteObject
            - s3:ListMultipartUploadParts
            - s3:AbortMultipartUpload
          Effect: Allow
          Resource:
            - "*"
          Principal: "*"
    RouteTableIds:
      - Ref: RouteTable
    ServiceName:
      Fn::Join:
        - ''
        - - com.amazonaws.
          - Ref: AWS::Region
          - ".s3"
    VpcId:
      Ref: VPC
```

Wir empfehlen Ihnen, auch eine benutzerdefinierte Richtlinie zu erstellen, die nur Anfragen von Ihrer VPC den Zugriff auf Ihre S3-Buckets ermöglicht. Weitere Informationen finden Sie unter [Endpoints for Amazon S3 \(Endpunkte für Amazon S3\)](#) im Amazon-VPC-Benutzerhandbuch.

Die folgende Richtlinie ermöglicht den Zugriff auf S3-Buckets. Bearbeiten Sie diese Richtlinie, um nur den Zugriff auf die Ressourcen zu ermöglichen, die Ihr Job benötigt.

```
{
  "Version": "2008-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": "*",
      "Action": [
        "s3:GetObject",
        "s3:PutObject",
        "s3:ListBucket",
        "s3:GetBucketLocation",
        "s3>DeleteObject",
        "s3:ListMultipartUploadParts",
        "s3:AbortMultipartUpload"
      ],
      "Resource": "*"
    }
  ]
}
```

Verwenden Sie Standard-DNS-Einstellungen für Ihre Endpunkt-Routentabelle, so dass Standard-URLs von Amazon S3 (z. B. <http://s3-aws-region.amazonaws.com/DOC-EXAMPLE-BUCKET>) aufgelöst werden. Wenn Sie keine Standard-DNS-Einstellungen verwenden, stellen Sie sicher, dass die URLs, mit denen Sie die Speicherorte der Daten in Ihren Jobs angeben, aufgelöst werden, indem Sie die Routing-Tabellen der Endpunkte konfigurieren. Informationen über Routing-Tabellen für VPC-Endpunkte finden Sie unter [Routing für Gateway-Endpunkte](#) im Amazon VPC Benutzerhandbuch.

Die standardmäßige Endpunktrichtlinie ermöglicht es Benutzern, Pakete aus den Amazon Linux- und Amazon Linux 2-Repositorys in unserem Job-Container zu installieren. Wenn Sie nicht möchten, dass Benutzer Pakete von diesem Repository installieren, erstellen Sie eine benutzerdefinierte Endpunkt-Richtlinie, die ausdrücklich den Zugriff auf die Amazon Linux- und Amazon Linux-2-Repositorys verweigert. Comprehend selbst benötigt keine solchen Pakete, sodass es keine Auswirkungen auf die Funktionalität gibt. Nachfolgend finden Sie eine Beispielrichtlinie, die den Zugriff auf diese Repositorys verweigert:

```
{
  "Statement": [
    {
```

```

        "Sid": "AmazonLinuxAMIRepositoryAccess",
        "Principal": "*",
        "Action": [
            "s3:GetObject"
        ],
        "Effect": "Deny",
        "Resource": [
            "arn:aws:s3:::packages.*.amazonaws.com/*",
            "arn:aws:s3:::repo.*.amazonaws.com/*"
        ]
    }
]
}
{
    "Statement": [
        {
            "Sid": "AmazonLinux2AMIRepositoryAccess",
            "Principal": "*",
            "Action": [
                "s3:GetObject"
            ],
            "Effect": "Deny",
            "Resource": [
                "arn:aws:s3:::amazonlinux.*.amazonaws.com/*"
            ]
        }
    ]
}
}

```

## Berechtigungen für **DataAccessRole**

Wenn Sie eine VPC mit Ihrem Analysejob verwenden, müssen die für die `Create*` und `DataAccessRole` verwendeten `Start*` Operationen auch über Berechtigungen für die VPC verfügen, von der aus auf die Eingabedokumente und den Ausgabe-Bucket zugegriffen wird.

Die folgende Richtlinie bietet den erforderlichen Zugriff auf die für `Create*` und `DataAccessRole` `Start*` verwendeten Operationen.

```

{
    "Version": "2008-10-17",
    "Statement": [
        {
            "Effect": "Allow",

```

```

    "Action": [
      "ec2:CreateNetworkInterface",
      "ec2:CreateNetworkInterfacePermission",
      "ec2>DeleteNetworkInterface",
      "ec2>DeleteNetworkInterfacePermission",
      "ec2:DescribeNetworkInterfaces",
      "ec2:DescribeVpcs",
      "ec2:DescribeDhcpOptions",
      "ec2:DescribeSubnets",
      "ec2:DescribeSecurityGroups"
    ],
    "Resource": "*"
  }
}

```

## Konfiguration der VPC-Sicherheitsgruppe

Bei verteilten Aufträgen müssen Sie die Kommunikation zwischen den verschiedenen Job-Containern im selben Job zulassen. Konfigurieren Sie dazu eine Regel für Ihre Sicherheitsgruppe, mit der eingehende Verbindungen zwischen Mitgliedern derselben Sicherheitsgruppe zugelassen werden. Weitere Informationen finden Sie unter [Sicherheitsgruppenregeln](#) im Amazon VPC-Benutzerhandbuch.

## Connect zu Ressourcen außerhalb Ihrer VPC her

Wenn Sie Ihre VPC so konfigurieren, dass sie keinen Internetzugang hat, haben Jobs, die diese VPC verwenden, keinen Zugriff auf Ressourcen außerhalb Ihrer VPC. Wenn Ihre Jobs Zugriff auf Ressourcen außerhalb Ihrer VPC benötigen, bieten Sie Zugriff mit einer der folgenden Optionen:

- Wenn Ihr Job Zugriff auf einen AWS Dienst benötigt, der VPC-Schnittstellen-Endpunkte unterstützt, erstellen Sie einen Endpunkt, um eine Verbindung zu diesem Dienst herzustellen. Eine Liste der Services, die Schnittstellenendpunkte unterstützen, finden Sie unter [VPC-Endpunkte](#) im Amazon VPC Benutzerhandbuch. Informationen zum Erstellen eines Schnittstellen-VPC-Endpunkts finden Sie unter [Interface VPC Endpoints \(AWS PrivateLink\)](#) im Amazon VPC-Benutzerhandbuch.
- Wenn Ihr Job Zugriff auf einen AWS Dienst benötigt, der keine VPC-Schnittstellen-Endpunkte unterstützt, oder auf eine Ressource außerhalb von AWS, erstellen Sie ein NAT-Gateway und konfigurieren Sie Ihre Sicherheitsgruppen so, dass ausgehende Verbindungen zugelassen werden. Informationen zur Einrichtung eines NAT-Gateways für Ihre VPC finden Sie unter [Szenario 2: VPC with Public and Private Subnets \(NAT\)](#) im Amazon VPC-Benutzerhandbuch.



## Amazon Comprehend und Schnittstellen-VPC-Endpunkte (AWS PrivateLink)

Sie können eine private Verbindung zwischen Ihrer VPC und Amazon Comprehend herstellen, indem Sie einen Schnittstellen-VPC-Endpunkt erstellen. Schnittstellenendpunkte werden von [unterstützt AWS PrivateLink](#), einer Technologie, mit der Sie ohne Internet-Gateway, NAT-Gerät, VPN-Verbindung oder AWS Direct-Connect-Verbindung privat auf Amazon Comprehend-APIs zugreifen können. Instances in Ihrer VPC benötigen für die Kommunikation mit Amazon Comprehend APIs keine öffentlichen IP-Adressen. Der Datenverkehr zwischen Ihrer VPC und Amazon Comprehend verlässt das Amazon-Netzwerk nicht.

Jeder Schnittstellenendpunkt wird durch eine oder mehrere [Elastic-Netzwerk-Schnittstellen](#) in Ihren Subnetzen dargestellt.

Weitere Informationen finden Sie unter [Schnittstellen-VPC-Endpunkte \(AWS PrivateLink\)](#) im Amazon-VPC-Benutzerhandbuch.

### Überlegungen zu Amazon Comprehend VPC-Endpunkten

Bevor Sie einen Schnittstellen-VPC-Endpunkt für Amazon Comprehend einrichten, lesen Sie die [Eigenschaften und Einschränkungen von Schnittstellenendpunkten](#) im Amazon-VPC-Benutzerhandbuch.

Amazon Comprehend-Endpunkte sind nicht in allen Availability Zones einer Region verfügbar. Verwenden Sie beim Erstellen des Endpunkts den folgenden Befehl, um die Availability Zones aufzulisten.

```
aws ec2 describe-vpc-endpoint-services \
  --service-names com.amazonaws.us-west-2.comprehend
```

Amazon Comprehend unterstützt Aufrufe aller API-Aktionen von Ihrer VPC aus.

### Erstellen eines Schnittstellen-VPC-Endpunkts für Amazon Comprehend

Sie können einen VPC-Endpunkt für den Amazon Comprehend-Service entweder über die Amazon VPC-Konsole oder die AWS Command Line Interface (AWS CLI) erstellen. Weitere Informationen finden Sie unter [Erstellung eines Schnittstellenendpunkts](#) im Benutzerhandbuch für Amazon VPC.

Erstellen Sie einen VPC-Endpunkt für Amazon Comprehend mit dem folgenden Servicenamen:

- `com.amazonaws.region.comprehend`

Wenn Sie ein privates DNS für den Endpunkt aktivieren, können Sie API-Anforderungen an Amazon Comprehend unter Verwendung seines standardmäßigen DNS-Namens für die Region senden, z. B. `comprehend.us-east-1.amazonaws.com`.

Weitere Informationen finden Sie unter [Zugriff auf einen Service über einen Schnittstellenendpunkt](#) im Benutzerhandbuch für Amazon VPC.

## Erstellen einer VPC-Endpunktrichtlinie für Amazon Comprehend

Sie können Ihrem VPC-Endpunkt eine Endpunktrichtlinie anfügen, die den Zugriff auf Amazon Comprehend steuert. Die Richtlinie gibt die folgenden Informationen an:

- Prinzipal, der die Aktionen ausführen kann.
- Aktionen, die ausgeführt werden können
- Die Ressourcen, für die Aktionen ausgeführt werden können.

Weitere Informationen finden Sie unter [Steuerung des Zugriffs auf Services mit VPC-Endpunkten](#) im Amazon-VPC-Benutzerhandbuch.

### Beispiel: VPC-Endpunktrichtlinie für Amazon Comprehend-Aktionen

Im Folgenden finden Sie ein Beispiel für eine Endpunktrichtlinie für Amazon Comprehend . Wenn diese Richtlinie an einen Endpunkt angefügt wird, gewährt sie allen Prinzipalen auf allen Ressourcen Zugriff auf die Amazon Comprehend-DetectEntitiesAktion.

```
{
  "Statement": [
    {
      "Principal": "*",
      "Effect": "Allow",
      "Action": [
        "comprehend:DetectEntities"
      ],
      "Resource": "*"
    }
  ]
}
```

# Identity and Access Management für Amazon Comprehend

AWS Identity and Access Management (IAM) hilft einem Administrator AWS-Service, den Zugriff auf Ressourcen sicher zu kontrollieren. AWS IAM-Administratoren kontrollieren, wer authentifiziert (angemeldet) und autorisiert werden kann (über Berechtigungen verfügt), um Amazon Comprehend Comprehend-Ressourcen zu verwenden. IAM ist ein Programm AWS-Service, das Sie ohne zusätzliche Kosten nutzen können.

## Themen

- [Zielgruppe](#)
- [Authentifizierung mit Identitäten](#)
- [Verwalten des Zugriffs mit Richtlinien](#)
- [So funktioniert Amazon Comprehend mit IAM](#)
- [Beispiele für identitätsbasierte Richtlinien für Amazon Comprehend](#)
- [AWS Von verwaltete Richtlinien für Amazon Comprehend](#)
- [Fehlerbehebung bei Amazon Comprehend Identity and Access](#)

## Zielgruppe

Wie Sie AWS Identity and Access Management (IAM) verwenden, hängt von der Arbeit ab, die Sie in Amazon Comprehend ausführen.

**Servicebenutzer** — Wenn Sie den Amazon Comprehend-Service für Ihre Arbeit verwenden, stellt Ihnen Ihr Administrator die Anmeldeinformationen und Berechtigungen zur Verfügung, die Sie benötigen. Da Sie für Ihre Arbeit mehr Funktionen von Amazon Comprehend verwenden, benötigen Sie möglicherweise zusätzliche Berechtigungen. Wenn Sie die Funktionsweise der Zugriffskontrolle nachvollziehen, wissen Sie bereits, welche Berechtigungen Sie von Ihrem Administrator anfordern müssen. Wenn Sie auf eine Funktion in Amazon Comprehend nicht zugreifen können, finden Sie weitere Informationen unter [Fehlerbehebung bei Amazon Comprehend Identity and Access](#)

**Service-Administrator** — Wenn Sie in Ihrem Unternehmen für die Amazon Comprehend-Ressourcen verantwortlich sind, haben Sie wahrscheinlich vollen Zugriff auf Amazon Comprehend. Es ist Ihre Aufgabe, zu bestimmen, auf welche Funktionen und Ressourcen von Amazon Comprehend Ihre Servicebenutzer zugreifen sollen. Sie müssen dann Anträge an Ihren IAM-Administrator stellen, um die Berechtigungen Ihrer Servicenutzer zu ändern. Lesen Sie die Informationen auf dieser Seite, um die Grundkonzepte von IAM nachzuvollziehen. Weitere Informationen darüber, wie Ihr

Unternehmen IAM mit Amazon Comprehend verwenden kann, finden Sie unter [So funktioniert Amazon Comprehend mit IAM](#)

IAM-Administrator — Wenn Sie ein IAM-Administrator sind, möchten Sie vielleicht mehr darüber erfahren, wie Sie Richtlinien schreiben können, um den Zugriff auf Amazon Comprehend zu verwalten. Beispiele für identitätsbasierte Amazon Comprehend Richtlinien, die Sie in IAM verwenden können, finden Sie unter [Beispiele für identitätsbasierte Richtlinien für Amazon Comprehend](#)

## Authentifizierung mit Identitäten

Authentifizierung ist die Art und Weise, wie Sie sich mit Ihren Identitätsdaten anmelden. AWS Sie müssen als IAM-Benutzer authentifiziert (angemeldet AWS) sein oder eine IAM-Rolle annehmen. Root-Benutzer des AWS-Kontos

Sie können sich AWS als föderierte Identität anmelden, indem Sie Anmeldeinformationen verwenden, die über eine Identitätsquelle bereitgestellt wurden. AWS IAM Identity Center (IAM Identity Center) -Benutzer, die Single Sign-On-Authentifizierung Ihres Unternehmens und Ihre Google- oder Facebook-Anmeldeinformationen sind Beispiele für föderierte Identitäten. Wenn Sie sich als Verbundidentität anmelden, hat der Administrator vorher mithilfe von IAM-Rollen einen Identitätsverbund eingerichtet. Wenn Sie über den Verbund darauf zugreifen AWS , übernehmen Sie indirekt eine Rolle.

Je nachdem, welcher Benutzertyp Sie sind, können Sie sich beim AWS Management Console oder beim AWS Zugangsportale anmelden. Weitere Informationen zur Anmeldung finden Sie AWS unter [So melden Sie sich bei Ihrem an AWS-Konto](#) im AWS-Anmeldung Benutzerhandbuch.

Wenn Sie AWS programmgesteuert zugreifen, AWS stellt es ein Software Development Kit (SDK) und eine Befehlszeilenschnittstelle (CLI) bereit, um Ihre Anfragen mithilfe Ihrer Anmeldeinformationen kryptografisch zu signieren. Wenn Sie keine AWS Tools verwenden, müssen Sie Anfragen selbst signieren. Weitere Informationen zur Verwendung der empfohlenen Methode, um Anfragen selbst zu [signieren, finden Sie im IAM-Benutzerhandbuch unter AWS API-Anfragen](#) signieren.

Unabhängig von der verwendeten Authentifizierungsmethode müssen Sie möglicherweise zusätzliche Sicherheitsinformationen angeben. AWS empfiehlt beispielsweise, die Multi-Faktor-Authentifizierung (MFA) zu verwenden, um die Sicherheit Ihres Kontos zu erhöhen. Weitere Informationen finden Sie unter [Multi-Faktor-Authentifizierung](#) im AWS IAM Identity Center - Benutzerhandbuch und [Verwenden der Multi-Faktor-Authentifizierung \(MFA\) in AWS](#) im IAM-Benutzerhandbuch.

## AWS-Konto Root-Benutzer

Wenn Sie ein AWS-Konto erstellen, beginnen Sie mit einer Anmeldeidentität, die vollständigen Zugriff auf alle AWS-Services Ressourcen im Konto hat. Diese Identität wird als AWS-Konto Root-Benutzer bezeichnet. Sie können darauf zugreifen, indem Sie sich mit der E-Mail-Adresse und dem Passwort anmelden, mit denen Sie das Konto erstellt haben. Wir raten ausdrücklich davon ab, den Root-Benutzer für Alltagsaufgaben zu verwenden. Schützen Sie Ihre Root-Benutzer-Anmeldeinformationen und verwenden Sie diese, um die Aufgaben auszuführen, die nur der Root-Benutzer ausführen kann. Eine vollständige Liste der Aufgaben, für die Sie sich als Root-Benutzer anmelden müssen, finden Sie unter [Aufgaben, die Root-Benutzer-Anmeldeinformationen erfordern](#) im IAM-Benutzerhandbuch.

## Verbundidentität

Als bewährte Methode sollten menschliche Benutzer, einschließlich Benutzer, die Administratorzugriff benötigen, für den Zugriff AWS-Services mithilfe temporärer Anmeldeinformationen den Verbund mit einem Identitätsanbieter verwenden.

Eine föderierte Identität ist ein Benutzer aus Ihrem Unternehmensbenutzerverzeichnis, einem Web-Identitätsanbieter AWS Directory Service, dem Identity Center-Verzeichnis oder einem beliebigen Benutzer, der mithilfe AWS-Services von Anmeldeinformationen zugreift, die über eine Identitätsquelle bereitgestellt wurden. Wenn föderierte Identitäten darauf zugreifen AWS-Konten, übernehmen sie Rollen, und die Rollen stellen temporäre Anmeldeinformationen bereit.

Für die zentrale Zugriffsverwaltung empfehlen wir Ihnen, AWS IAM Identity Center zu verwenden. Sie können Benutzer und Gruppen in IAM Identity Center erstellen, oder Sie können eine Verbindung zu einer Gruppe von Benutzern und Gruppen in Ihrer eigenen Identitätsquelle herstellen und diese synchronisieren, um sie in all Ihren AWS-Konten Anwendungen zu verwenden. Informationen zu IAM Identity Center finden Sie unter [Was ist IAM Identity Center?](#) im AWS IAM Identity Center - Benutzerhandbuch.

## IAM-Benutzer und -Gruppen

Ein [IAM-Benutzer](#) ist eine Identität innerhalb Ihres Unternehmens AWS-Konto, die über spezifische Berechtigungen für eine einzelne Person oder Anwendung verfügt. Wenn möglich, empfehlen wir, temporäre Anmeldeinformationen zu verwenden, anstatt IAM-Benutzer zu erstellen, die langfristige Anmeldeinformationen wie Passwörter und Zugriffsschlüssel haben. Bei speziellen Anwendungsfällen, die langfristige Anmeldeinformationen mit IAM-Benutzern erfordern, empfehlen wir jedoch, die Zugriffsschlüssel zu rotieren. Weitere Informationen finden Sie unter [Regelmäßiges](#)

[Rotieren von Zugriffsschlüsseln für Anwendungsfälle, die langfristige Anmeldeinformationen erfordern](#) im IAM-Benutzerhandbuch.

Eine [IAM-Gruppe](#) ist eine Identität, die eine Sammlung von IAM-Benutzern angibt. Sie können sich nicht als Gruppe anmelden. Mithilfe von Gruppen können Sie Berechtigungen für mehrere Benutzer gleichzeitig angeben. Gruppen vereinfachen die Verwaltung von Berechtigungen, wenn es zahlreiche Benutzer gibt. Sie könnten beispielsweise einer Gruppe mit dem Namen IAMAdmins Berechtigungen zum Verwalten von IAM-Ressourcen erteilen.

Benutzer unterscheiden sich von Rollen. Ein Benutzer ist einer einzigen Person oder Anwendung eindeutig zugeordnet. Eine Rolle kann von allen Personen angenommen werden, die sie benötigen. Benutzer besitzen dauerhafte Anmeldeinformationen. Rollen stellen temporäre Anmeldeinformationen bereit. Weitere Informationen finden Sie unter [Erstellen eines IAM-Benutzers \(anstatt einer Rolle\)](#) im IAM-Benutzerhandbuch.

## IAM-Rollen

Eine [IAM-Rolle](#) ist eine Identität innerhalb Ihres Unternehmens AWS-Konto , die über bestimmte Berechtigungen verfügt. Sie ist einem IAM-Benutzer vergleichbar, ist aber nicht mit einer bestimmten Person verknüpft. Sie können vorübergehend eine IAM-Rolle in der übernehmen, AWS Management Console indem Sie die Rollen [wechseln](#). Sie können eine Rolle übernehmen, indem Sie eine AWS CLI oder AWS API-Operation aufrufen oder eine benutzerdefinierte URL verwenden. Weitere Informationen zu Methoden für die Verwendung von Rollen finden Sie unter [Verwenden von IAM-Rollen](#) im IAM-Benutzerhandbuch.

IAM-Rollen mit temporären Anmeldeinformationen sind in folgenden Situationen hilfreich:

- **Verbundbenutzerzugriff** – Um einer Verbundidentität Berechtigungen zuzuweisen, erstellen Sie eine Rolle und definieren Berechtigungen für die Rolle. Wird eine Verbundidentität authentifiziert, so wird die Identität der Rolle zugeordnet und erhält die von der Rolle definierten Berechtigungen. Informationen zu Rollen für den Verbund finden Sie unter [Erstellen von Rollen für externe Identitätsanbieter](#) im IAM-Benutzerhandbuch. Wenn Sie IAM Identity Center verwenden, konfigurieren Sie einen Berechtigungssatz. Wenn Sie steuern möchten, worauf Ihre Identitäten nach der Authentifizierung zugreifen können, korreliert IAM Identity Center den Berechtigungssatz mit einer Rolle in IAM. Informationen zu Berechtigungssätzen finden Sie unter [Berechtigungssätze](#) im AWS IAM Identity Center -Benutzerhandbuch.
- **Temporäre IAM-Benutzerberechtigungen** – Ein IAM-Benutzer oder eine -Rolle kann eine IAM-Rolle übernehmen, um vorübergehend andere Berechtigungen für eine bestimmte Aufgabe zu erhalten.

- **Kontoübergreifender Zugriff** – Sie können eine IAM-Rolle verwenden, um einem vertrauenswürdigen Prinzipal in einem anderen Konto den Zugriff auf Ressourcen in Ihrem Konto zu ermöglichen. Rollen stellen die primäre Möglichkeit dar, um kontoübergreifendem Zugriff zu gewähren. Bei einigen können Sie AWS-Services jedoch eine Richtlinie direkt an eine Ressource anhängen (anstatt eine Rolle als Proxy zu verwenden). Informationen zum Unterschied zwischen Rollen und ressourcenbasierten Richtlinien für den kontoübergreifenden Zugriff finden Sie unter [Kontoübergreifender Ressourcenzugriff in IAM im IAM-Benutzerhandbuch](#).
- **Serviceübergreifender Zugriff** — Einige verwenden Funktionen in anderen. AWS-Services AWS-Services Wenn Sie beispielsweise einen Aufruf in einem Service tätigen, führt dieser Service häufig Anwendungen in Amazon-EC2 aus oder speichert Objekte in Amazon-S3. Ein Dienst kann dies mit den Berechtigungen des aufrufenden Prinzipals mit einer Servicerolle oder mit einer serviceverknüpften Rolle tun.
- **Forward Access Sessions (FAS)** — Wenn Sie einen IAM-Benutzer oder eine IAM-Rolle verwenden, um Aktionen auszuführen AWS, gelten Sie als Principal. Bei einigen Services könnte es Aktionen geben, die dann eine andere Aktion in einem anderen Service initiieren. FAS verwendet die Berechtigungen des Prinzipals, der einen aufruft AWS-Service, in Kombination mit der Anfrage, Anfragen an AWS-Service nachgelagerte Dienste zu stellen. FAS-Anfragen werden nur gestellt, wenn ein Dienst eine Anfrage erhält, für deren Abschluss Interaktionen mit anderen AWS-Services oder Ressourcen erforderlich sind. In diesem Fall müssen Sie über Berechtigungen zum Ausführen beider Aktionen verfügen. Einzelheiten zu den Richtlinien für FAS-Anfragen finden Sie unter [Zugriffssitzungen weiterleiten](#).
- **Servicerolle** – Eine Servicerolle ist eine [IAM-Rolle](#), die ein Service übernimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM-Administrator kann eine Servicerolle innerhalb von IAM erstellen, ändern und löschen. Weitere Informationen finden Sie unter [Erstellen einer Rolle zum Delegieren von Berechtigungen an einen AWS-Service](#) im IAM-Benutzerhandbuch.
- **Dienstbezogene Rolle** — Eine dienstbezogene Rolle ist eine Art von Servicerolle, die mit einer verknüpft ist. AWS-Service Der Service kann die Rolle übernehmen, um eine Aktion in Ihrem Namen auszuführen. Servicebezogene Rollen erscheinen in Ihrem Dienst AWS-Konto und gehören dem Dienst. Ein IAM-Administrator kann die Berechtigungen für Service-verknüpfte Rollen anzeigen, aber nicht bearbeiten.
- **Auf Amazon EC2 ausgeführte Anwendungen** — Sie können eine IAM-Rolle verwenden, um temporäre Anmeldeinformationen für Anwendungen zu verwalten, die auf einer EC2-Instance ausgeführt werden und API-Anfragen stellen AWS CLI . AWS Das ist eher zu empfehlen, als Zugriffsschlüssel innerhalb der EC2-Instance zu speichern. Um einer EC2-Instance eine AWS Rolle zuzuweisen und sie allen ihren Anwendungen zur Verfügung zu stellen, erstellen Sie

ein Instance-Profil, das an die Instance angehängt ist. Ein Instance-Profil enthält die Rolle und ermöglicht, dass Programme, die in der EC2-Instance ausgeführt werden, temporäre Anmeldeinformationen erhalten. Weitere Informationen finden Sie unter [Verwenden einer IAM-Rolle zum Erteilen von Berechtigungen für Anwendungen, die auf Amazon-EC2-Instances ausgeführt werden](#) im IAM-Benutzerhandbuch.

Informationen dazu, wann Sie IAM-Rollen oder IAM-Benutzer verwenden sollten, finden Sie unter [Erstellen einer IAM-Rolle \(anstatt eines Benutzers\)](#) im IAM-Benutzerhandbuch.

## Verwalten des Zugriffs mit Richtlinien

Sie kontrollieren den Zugriff, AWS indem Sie Richtlinien erstellen und diese an AWS Identitäten oder Ressourcen anhängen. Eine Richtlinie ist ein Objekt, AWS das, wenn es einer Identität oder Ressource zugeordnet ist, deren Berechtigungen definiert. AWS wertet diese Richtlinien aus, wenn ein Prinzipal (Benutzer, Root-Benutzer oder Rollensitzung) eine Anfrage stellt. Berechtigungen in den Richtlinien bestimmen, ob die Anforderung zugelassen oder abgelehnt wird. Die meisten Richtlinien werden AWS als JSON-Dokumente gespeichert. Weitere Informationen zu Struktur und Inhalten von JSON-Richtliniendokumenten finden Sie unter [Übersicht über JSON-Richtlinien](#) im IAM-Benutzerhandbuch.

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer auf was Zugriff hat. Das bedeutet, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Standardmäßig haben Benutzer, Gruppen und Rollen keine Berechtigungen. Ein IAM-Administrator muss IAM-Richtlinien erstellen, die Benutzern die Berechtigung erteilen, Aktionen für die Ressourcen auszuführen, die sie benötigen. Der Administrator kann dann die IAM-Richtlinien zu Rollen hinzufügen, und Benutzer können die Rollen annehmen.

IAM-Richtlinien definieren Berechtigungen für eine Aktion unabhängig von der Methode, die Sie zur Ausführung der Aktion verwenden. Angenommen, es gibt eine Richtlinie, die Berechtigungen für die `iam:GetRole`-Aktion erteilt. Ein Benutzer mit dieser Richtlinie kann Rolleninformationen von der AWS Management Console AWS CLI, der oder der AWS API abrufen.

## Identitätsbasierte Richtlinien

Identitätsbasierte Richtlinien sind JSON-Berechtigungsrichtliniendokumente, die Sie einer Identität anfügen können, wie z. B. IAM-Benutzern, -Benutzergruppen oder -Rollen. Diese Richtlinien steuern,



welche Aktionen die Benutzer und Rollen für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter [Erstellen von IAM-Richtlinien](#) im IAM-Benutzerhandbuch.

Identitätsbasierte Richtlinien können weiter als Inline-Richtlinien oder verwaltete Richtlinien kategorisiert werden. Inline-Richtlinien sind direkt in einen einzelnen Benutzer, eine einzelne Gruppe oder eine einzelne Rolle eingebettet. Verwaltete Richtlinien sind eigenständige Richtlinien, die Sie mehreren Benutzern, Gruppen und Rollen in Ihrem System zuordnen können AWS-Konto. Zu den verwalteten Richtlinien gehören AWS verwaltete Richtlinien und vom Kunden verwaltete Richtlinien. Informationen dazu, wie Sie zwischen einer verwalteten Richtlinie und einer eingebundenen Richtlinie wählen, finden Sie unter [Auswahl zwischen verwalteten und eingebundenen Richtlinien](#) im IAM-Benutzerhandbuch.

## Ressourcenbasierte Richtlinien

Ressourcenbasierte Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anfügen. Beispiele für ressourcenbasierte Richtlinien sind IAM-Rollen-Vertrauensrichtlinien und Amazon-S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Für die Ressource, an welche die Richtlinie angehängt ist, legt die Richtlinie fest, welche Aktionen ein bestimmter Prinzipal unter welchen Bedingungen für diese Ressource ausführen kann. Sie müssen in einer ressourcenbasierten Richtlinie [einen Prinzipal angeben](#). Zu den Prinzipalen können Konten, Benutzer, Rollen, Verbundbenutzer oder gehören. AWS-Services

Ressourcenbasierte Richtlinien sind Richtlinien innerhalb dieses Diensts. Sie können AWS verwaltete Richtlinien von IAM nicht in einer ressourcenbasierten Richtlinie verwenden.

## Zugriffssteuerungslisten (ACLs)

Zugriffssteuerungslisten (ACLs) steuern, welche Prinzipale (Kontomitglieder, Benutzer oder Rollen) auf eine Ressource zugreifen können. ACLs sind ähnlich wie ressourcenbasierte Richtlinien, verwenden jedoch nicht das JSON-Richtliniendokumentformat.

Amazon S3 und Amazon VPC sind Beispiele für Services, die ACLs unterstützen. AWS WAF Weitere Informationen“ zu ACLs finden Sie unter [Zugriffskontrollliste \(ACL\) – Übersicht](#) (Access Control List) im Amazon-Simple-Storage-Service-Entwicklerhandbuch.

## Weitere Richtlinientypen

AWS unterstützt zusätzliche, weniger verbreitete Richtlinientypen. Diese Richtlinientypen können die maximalen Berechtigungen festlegen, die Ihnen von den häufiger verwendeten Richtlinientypen erteilt werden können.

- **Berechtigungsgrenzen** – Eine Berechtigungsgrenze ist ein erweitertes Feature, mit der Sie die maximalen Berechtigungen festlegen können, die eine identitätsbasierte Richtlinie einer IAM-Entität (IAM-Benutzer oder -Rolle) erteilen kann. Sie können eine Berechtigungsgrenze für eine Entität festlegen. Die daraus resultierenden Berechtigungen sind der Schnittpunkt der identitätsbasierten Richtlinien einer Entität und ihrer Berechtigungsgrenzen. Ressourcenbasierte Richtlinien, die den Benutzer oder die Rolle im Feld `Principal` angeben, werden nicht durch Berechtigungsgrenzen eingeschränkt. Eine explizite Zugriffsverweigerung in einer dieser Richtlinien setzt eine Zugriffserlaubnis außer Kraft. Weitere Informationen über Berechtigungsgrenzen finden Sie unter [Berechtigungsgrenzen für IAM-Entitäten](#) im IAM-Benutzerhandbuch.
- **Service Control Policies (SCPs)** — SCPs sind JSON-Richtlinien, die die maximalen Berechtigungen für eine Organisation oder Organisationseinheit (OU) in festlegen. AWS Organizations AWS Organizations ist ein Dienst zur Gruppierung und zentralen Verwaltung mehrerer Objekte AWS-Konten , die Ihrem Unternehmen gehören. Wenn Sie innerhalb einer Organisation alle Features aktivieren, können Sie Service-Kontrollrichtlinien (SCPs) auf alle oder einzelne Ihrer Konten anwenden. Das SCP schränkt die Berechtigungen für Entitäten in Mitgliedskonten ein, einschließlich der einzelnen Entitäten. Root-Benutzer des AWS-Kontos Weitere Informationen zu Organizations und SCPs finden Sie unter [Funktionsweise von SCPs](#) im AWS Organizations -Benutzerhandbuch.
- **Sitzungsrichtlinien** – Sitzungsrichtlinien sind erweiterte Richtlinien, die Sie als Parameter übergeben, wenn Sie eine temporäre Sitzung für eine Rolle oder einen verbundenen Benutzer programmgesteuert erstellen. Die resultierenden Sitzungsberechtigungen sind eine Schnittmenge der auf der Identität des Benutzers oder der Rolle basierenden Richtlinien und der Sitzungsrichtlinien. Berechtigungen können auch aus einer ressourcenbasierten Richtlinie stammen. Eine explizite Zugriffsverweigerung in einer dieser Richtlinien setzt eine Zugriffserlaubnis außer Kraft. Weitere Informationen finden Sie unter [Sitzungsrichtlinien](#) im IAM-Benutzerhandbuch.

## Mehrere Richtlinientypen

Wenn mehrere auf eine Anforderung mehrere Richtlinientypen angewendet werden können, sind die entsprechenden Berechtigungen komplizierter. Informationen darüber, wie AWS bestimmt wird,

ob eine Anfrage zulässig ist, wenn mehrere Richtlinientypen betroffen sind, finden Sie im IAM-Benutzerhandbuch unter [Bewertungslogik für Richtlinien](#).

## So funktioniert Amazon Comprehend mit IAM

Bevor Sie IAM verwenden, um den Zugriff auf Amazon Comprehend zu verwalten, sollten Sie sich darüber informieren, welche IAM-Funktionen mit Amazon Comprehend verwendet werden können.

IAM-Funktionen, die Sie mit Amazon Comprehend verwenden können

IAM-Feature	Unterstützung von Amazon Comprehend
<a href="#">Identitätsbasierte Richtlinien</a>	Ja
<a href="#">Ressourcenbasierte Richtlinien</a>	Ja
<a href="#">Richtlinienaktionen</a>	Ja
<a href="#">Richtlinienressourcen</a>	Ja
<a href="#">Richtlinienbedingungsschlüssel (servicespezifisch)</a>	Ja
<a href="#">ACLs</a>	Nein
<a href="#">ABAC (Tags in Richtlinien)</a>	Teilweise
<a href="#">Temporäre Anmeldeinformationen</a>	Ja
<a href="#">Forward Access Sessions (FAS)</a>	Ja
<a href="#">Servicerollen</a>	Ja
<a href="#">Service-verknüpfte Rollen</a>	Nein

Einen allgemeinen Überblick darüber, wie Amazon Comprehend und andere AWS Services mit den meisten IAM-Funktionen funktionieren, finden Sie im [AWS IAM-Benutzerhandbuch unter Services, die mit IAM funktionieren](#).

## Identitätsbasierte Richtlinien für Amazon Comprehend

Unterstützt Richtlinien auf Identitätsbasis. Ja

Identitätsbasierte Richtlinien sind JSON-Berechtigungsrichtliniendokumente, die Sie einer Identität anfügen können, wie z. B. IAM-Benutzern, -Benutzergruppen oder -Rollen. Diese Richtlinien steuern, welche Aktionen die Benutzer und Rollen für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter [Erstellen von IAM-Richtlinien](#) im IAM-Benutzerhandbuch.

Mit identitätsbasierten IAM-Richtlinien können Sie angeben, welche Aktionen und Ressourcen zugelassen oder abgelehnt werden. Darüber hinaus können Sie die Bedingungen festlegen, unter denen Aktionen zugelassen oder abgelehnt werden. Sie können den Prinzipal nicht in einer identitätsbasierten Richtlinie angeben, da er für den Benutzer oder die Rolle gilt, dem er zugeordnet ist. Informationen zu sämtlichen Elementen, die Sie in einer JSON-Richtlinie verwenden, finden Sie in der [IAM-Referenz für JSON-Richtlinienelemente](#) im IAM-Benutzerhandbuch.

### Beispiele für identitätsbasierte Richtlinien für Amazon Comprehend

Beispiele für identitätsbasierte Richtlinien von Amazon Comprehend finden Sie unter [Beispiele für identitätsbasierte Richtlinien für Amazon Comprehend](#)

## Ressourcenbasierte Richtlinien innerhalb von Amazon Comprehend

Unterstützt ressourcenbasierte Richtlinien Ja

Ressourcenbasierte Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anfügen. Beispiele für ressourcenbasierte Richtlinien sind IAM-Rollen-Vertrauensrichtlinien und Amazon-S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Für die Ressource, an welche die Richtlinie angehängt ist, legt die Richtlinie fest, welche Aktionen ein bestimmter Prinzipal unter welchen Bedingungen für diese Ressource ausführen kann. Sie müssen in einer ressourcenbasierten Richtlinie [einen Prinzipal angeben](#). Zu den Prinzipalen können Konten, Benutzer, Rollen, Verbundbenutzer oder gehören. AWS-Services

Um kontoübergreifenden Zugriff zu ermöglichen, können Sie ein gesamtes Konto oder IAM-Entitäten in einem anderen Konto als Prinzipal in einer ressourcenbasierten Richtlinie angeben. Durch das Hinzufügen eines kontoübergreifenden Auftraggebers zu einer ressourcenbasierten Richtlinie ist nur die halbe Vertrauensbeziehung eingerichtet. Wenn sich der Prinzipal und die Ressource unterscheiden AWS-Konten, muss ein IAM-Administrator des vertrauenswürdigen Kontos auch der Prinzipalentsität (Benutzer oder Rolle) die Berechtigung zum Zugriff auf die Ressource erteilen. Sie erteilen Berechtigungen, indem Sie der juristischen Stelle eine identitätsbasierte Richtlinie anfügen. Wenn jedoch eine ressourcenbasierte Richtlinie Zugriff auf einen Prinzipal in demselben Konto gewährt, ist keine zusätzliche identitätsbasierte Richtlinie erforderlich. Weitere Informationen finden Sie unter [Kontenübergreifender Ressourcenzugriff in IAM](#) im IAM-Benutzerhandbuch.

Der Amazon Comprehend-Service unterstützt nur eine Art von ressourcenbasierter Richtlinie (eine benutzerdefinierte Modellrichtlinie), die an ein benutzerdefiniertes Modell angehängt ist. Diese Richtlinie definiert andere Konten, die das benutzerdefinierte Modell verwenden können.

Informationen zum Anhängen einer ressourcenbasierten Richtlinie an ein benutzerdefiniertes Modell finden Sie unter [Ressourcenbasierte Richtlinien für benutzerdefinierte Modelle](#)

## Politische Maßnahmen für Amazon Comprehend

Unterstützt Richtlinienaktionen	Ja
---------------------------------	----

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer Zugriff auf was hat. Das heißt, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das Element `Action` einer JSON-Richtlinie beschreibt die Aktionen, mit denen Sie den Zugriff in einer Richtlinie zulassen oder verweigern können. Richtlinienaktionen haben normalerweise denselben Namen wie der zugehörige AWS API-Vorgang. Es gibt einige Ausnahmen, z. B. Aktionen, die nur mit Genehmigung durchgeführt werden können und für die es keinen passenden API-Vorgang gibt. Es gibt auch einige Operationen, die mehrere Aktionen in einer Richtlinie erfordern. Diese zusätzlichen Aktionen werden als abhängige Aktionen bezeichnet.

Schließen Sie Aktionen in eine Richtlinie ein, um Berechtigungen zur Durchführung der zugeordneten Operation zu erteilen.

Eine Liste der Amazon Comprehend-Aktionen finden Sie unter [Von Amazon Comprehend definierte Aktionen in der Service Authorization Reference](#).

Richtlinienaktionen in Amazon Comprehend verwenden das folgende Präfix vor der Aktion:

```
comprehend
```

Um mehrere Aktionen in einer einzigen Anweisung anzugeben, trennen Sie sie mit Kommata:

```
"Action": [  
  "comprehend:DetectSentiment",  
  "comprehend:ClassifyDocument"  
]
```

Sie können auch Platzhalter verwenden, um mehrere Aktionen anzugeben. Beispielsweise können Sie alle Aktionen festlegen, die mit dem Wort `Describe` beginnen, einschließlich der folgenden Aktion:

```
"Action": "comprehend:Describe*"
```

Verwenden Sie keine Platzhalter, um alle Aktionen für einen Service anzugeben. Verwenden Sie die bewährte Methode der Gewährung der geringsten Rechte, wenn Sie die Berechtigungen in einer Richtlinie angeben.

Beispiele für identitätsbasierte Richtlinien von Amazon Comprehend finden Sie unter [Beispiele für identitätsbasierte Richtlinien für Amazon Comprehend](#)

## Richtlinienressourcen für Amazon Comprehend

Unterstützt Richtlinienressourcen

Ja

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer Zugriff auf was hat. Das bedeutet die Festlegung, welcher Prinzipal Aktionen für welche Ressourcen unter welchen Bedingungen ausführen kann.

Das JSON-Richtlinienelement `Resource` gibt die Objekte an, auf welche die Aktion angewendet wird. Anweisungen müssen entweder ein `Resource` oder ein `NotResource`-Element enthalten. Als bewährte Methode geben Sie eine Ressource mit dem zugehörigen [Amazon-Ressourcennamen](#)

[\(ARN\)](#) an. Sie können dies für Aktionen tun, die einen bestimmten Ressourcentyp unterstützen, der als Berechtigungen auf Ressourcenebene bezeichnet wird.

Verwenden Sie für Aktionen, die keine Berechtigungen auf Ressourcenebene unterstützen, z. B. Auflistungsoperationen, einen Platzhalter (\*), um anzugeben, dass die Anweisung für alle Ressourcen gilt.

```
"Resource": "*"
```

Eine Liste der Amazon Comprehend-Ressourcentypen und ihrer ARNs finden Sie unter [Von Amazon Comprehend definierte Ressourcen](#) in der Service Authorization Reference. Informationen darüber, mit welchen Aktionen Sie den ARN jeder Ressource angeben können, finden Sie unter [Von Amazon Comprehend definierte Aktionen](#).

## Schlüssel zu den Richtlinienbedingungen für Amazon Comprehend

Unterstützt servicespezifische Richtlinienbedingungsschlüssel	Ja
---	----

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer Zugriff auf was hat. Das heißt, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das Element `Condition` (oder `Condition block`) ermöglicht Ihnen die Angabe der Bedingungen, unter denen eine Anweisung wirksam ist. Das Element `Condition` ist optional. Sie können bedingte Ausdrücke erstellen, die [Bedingungsoperatoren](#) verwenden, z. B. `ist gleich` oder `kleiner als`, damit die Bedingung in der Richtlinie mit Werten in der Anforderung übereinstimmt.

Wenn Sie mehrere `Condition`-Elemente in einer Anweisung oder mehrere Schlüssel in einem einzelnen `Condition`-Element angeben, wertet AWS diese mittels einer logischen AND-Operation aus. Wenn Sie mehrere Werte für einen einzelnen Bedingungsschlüssel angeben, wertet die Bedingung mithilfe einer logischen OR Operation aus. Alle Bedingungen müssen erfüllt werden, bevor die Berechtigungen der Anweisung gewährt werden.

Sie können auch Platzhaltervariablen verwenden, wenn Sie Bedingungen angeben. Beispielsweise können Sie einem IAM-Benutzer die Berechtigung für den Zugriff auf eine Ressource nur dann

gewähren, wenn sie mit dessen IAM-Benutzernamen gekennzeichnet ist. Weitere Informationen finden Sie unter [IAM-Richtlinienelemente: Variablen und Tags](#) im IAM-Benutzerhandbuch.

AWS unterstützt globale Bedingungsschlüssel und dienstspezifische Bedingungsschlüssel. Eine Übersicht aller AWS globalen Bedingungsschlüssel finden Sie unter [Kontextschlüssel für AWS globale Bedingungen](#) im IAM-Benutzerhandbuch.

Eine Liste der Amazon Comprehend-Bedingungsschlüssel finden Sie unter [Bedingungsschlüssel für Amazon Comprehend](#) in der Service Authorization Reference. Informationen zu den Aktionen und Ressourcen, mit denen Sie einen Bedingungsschlüssel verwenden können, finden Sie unter [Von Amazon Comprehend definierte Aktionen](#).

Beispiele für identitätsbasierte Richtlinien von Amazon Comprehend finden Sie unter [Beispiele für identitätsbasierte Richtlinien für Amazon Comprehend](#)

## ACLs in Amazon Comprehend

Unterstützt ACLs

Nein

Zugriffssteuerungslisten (ACLs) steuern, welche Prinzipale (Kontomitglieder, Benutzer oder Rollen) auf eine Ressource zugreifen können. ACLs sind ähnlich wie ressourcenbasierte Richtlinien, verwenden jedoch nicht das JSON-Richtliniendokumentformat.

## ABAC mit Amazon Comprehend

Unterstützt ABAC (Tags in Richtlinien)

Teilweise

Die attributbasierte Zugriffskontrolle (ABAC) ist eine Autorisierungsstrategie, bei der Berechtigungen basierend auf Attributen definiert werden. In werden AWS diese Attribute als Tags bezeichnet. Sie können Tags an IAM-Entitäten (Benutzer oder Rollen) und an viele AWS Ressourcen anhängen. Das Markieren von Entitäten und Ressourcen ist der erste Schritt von ABAC. Anschließend entwerfen Sie ABAC-Richtlinien, um Operationen zuzulassen, wenn das Tag des Prinzipals mit dem Tag der Ressource übereinstimmt, auf die sie zugreifen möchten.

ABAC ist in Umgebungen hilfreich, die schnell wachsen, und unterstützt Sie in Situationen, in denen die Richtlinienverwaltung mühsam wird.



Um den Zugriff auf der Grundlage von Tags zu steuern, geben Sie im Bedingungelement einer [Richtlinie Tag-Informationen](#) an, indem Sie die Schlüssel `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, oder Bedingung `aws:TagKeys` verwenden.

Wenn ein Service alle drei Bedingungsschlüssel für jeden Ressourcentyp unterstützt, lautet der Wert für den Service Ja. Wenn ein Service alle drei Bedingungsschlüssel für nur einige Ressourcentypen unterstützt, lautet der Wert Teilweise.

Weitere Informationen zu ABAC finden Sie unter [Was ist ABAC?](#) im IAM-Benutzerhandbuch. Um ein Tutorial mit Schritten zur Einstellung von ABAC anzuzeigen, siehe [Attributbasierte Zugriffskontrolle \(ABAC\)](#) verwenden im IAM-Benutzerhandbuch.

Weitere Informationen zum Markieren von Amazon Comprehend Comprehend-Ressourcen finden Sie unter [Markieren Ihrer -Ressourcen](#)

## Temporäre Anmeldeinformationen mit Amazon Comprehend verwenden

Unterstützt temporäre Anmeldeinformationen	Ja
--	----

Einige funktionieren AWS-Services nicht, wenn Sie sich mit temporären Anmeldeinformationen anmelden. Weitere Informationen, einschließlich Informationen, die mit temporären Anmeldeinformationen AWS-Services [funktionieren AWS-Services](#) , [finden Sie im IAM-Benutzerhandbuch unter Diese Option funktioniert mit IAM](#).

Sie verwenden temporäre Anmeldeinformationen, wenn Sie sich mit einer anderen AWS Management Console Methode als einem Benutzernamen und einem Passwort anmelden. Wenn Sie beispielsweise AWS über den Single Sign-On-Link (SSO) Ihres Unternehmens darauf zugreifen, werden bei diesem Vorgang automatisch temporäre Anmeldeinformationen erstellt. Sie erstellen auch automatisch temporäre Anmeldeinformationen, wenn Sie sich als Benutzer bei der Konsole anmelden und dann die Rollen wechseln. Weitere Informationen zum Wechseln von Rollen finden Sie unter [Wechseln zu einer Rolle \(Konsole\)](#) im IAM-Benutzerhandbuch.

Mithilfe der AWS API AWS CLI oder können Sie temporäre Anmeldeinformationen manuell erstellen. Sie können diese temporären Anmeldeinformationen dann für den Zugriff verwenden AWS. AWS empfiehlt, temporäre Anmeldeinformationen dynamisch zu generieren, anstatt langfristige Zugriffsschlüssel zu verwenden. Weitere Informationen finden Sie unter [Temporäre Sicherheitsanmeldeinformationen in IAM](#).

## Zugriffssitzungen für Amazon Comprehend weiterleiten

Unterstützt Forward Access Sessions (FAS)	Ja
---	----

Wenn Sie einen IAM-Benutzer oder eine IAM-Rolle verwenden, um Aktionen auszuführen AWS, gelten Sie als Principal. Bei einigen Services könnte es Aktionen geben, die dann eine andere Aktion in einem anderen Service initiieren. FAS verwendet die Berechtigungen des Prinzipals, der einen aufruft AWS-Service, kombiniert mit der Anforderung, Anfragen an nachgelagerte Dienste AWS-Service zu stellen. FAS-Anfragen werden nur gestellt, wenn ein Dienst eine Anfrage erhält, für deren Abschluss Interaktionen mit anderen AWS-Services oder Ressourcen erforderlich sind. In diesem Fall müssen Sie über Berechtigungen zum Ausführen beider Aktionen verfügen. Einzelheiten zu den Richtlinien für FAS-Anfragen finden Sie unter [Zugriffssitzungen weiterleiten](#).

## Servicerollen für Amazon Comprehend

Unterstützt Servicerollen	Ja
---------------------------	----

Eine Servicerolle ist eine [IAM-Rolle](#), die ein Service annimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM-Administrator kann eine Servicerolle innerhalb von IAM erstellen, ändern und löschen. Weitere Informationen finden Sie unter [Erstellen einer Rolle zum Delegieren von Berechtigungen an einen AWS-Service](#) im IAM-Benutzerhandbuch.

### Warning

Das Ändern der Berechtigungen für eine Servicerolle kann die Funktionalität von Amazon Comprehend beeinträchtigen. Bearbeiten Sie Servicerollen nur, wenn Amazon Comprehend Sie dazu anleitet.

Um die asynchronen Amazon Comprehend-Operationen verwenden zu können, müssen Sie Amazon Comprehend Zugriff auf den Amazon S3 S3-Bucket gewähren, der Ihre Dokumentensammlung enthält. Dazu erstellen Sie in Ihrem Konto eine Datenzugriffsrolle mit einer Vertrauensrichtlinie, um dem Amazon Comprehend Service Principal zu vertrauen.

Ein Beispiel für eine Richtlinie finden Sie unter [Rollenbasierte Berechtigungen, die für asynchrone Operationen erforderlich sind](#)

## Servicebezogene Rollen für Amazon Comprehend

Unterstützt serviceverknüpfte Rollen

Nein

Eine serviceverknüpfte Rolle ist eine Art von Servicerolle, die mit einer verknüpft ist. AWS-Service Der Service kann die Rolle übernehmen, um eine Aktion in Ihrem Namen auszuführen. Dienstbezogene Rollen werden in Ihrem Dienst angezeigt AWS-Konto und gehören dem Dienst. Ein IAM-Administrator kann die Berechtigungen für Service-verknüpfte Rollen anzeigen, aber nicht bearbeiten.

Details zum Erstellen oder Verwalten von serviceverknüpften Rollen finden Sie unter [AWS -Services, die mit IAM funktionieren](#). Suchen Sie in der Tabelle nach einem Service mit einem Yes in der Spalte Service-linked role (Serviceverknüpfte Rolle). Wählen Sie den Link Yes (Ja) aus, um die Dokumentation für die serviceverknüpfte Rolle für diesen Service anzuzeigen.

## Beispiele für identitätsbasierte Richtlinien für Amazon Comprehend

Benutzer und Rollen besitzen standardmäßig keine Berechtigungen zum Erstellen oder Ändern von Amazon Comprehend-Ressourcen. Sie können auch keine Aufgaben über die AWS Management Console, die AWS Command Line Interface (AWS CLI) oder die AWS-API ausführen. Ein IAM-Administrator muss IAM-Richtlinien erstellen, die Benutzern die Berechtigung erteilen, Aktionen für die Ressourcen auszuführen, die sie benötigen. Der Administrator kann dann die IAM-Richtlinien zu Rollen hinzufügen, und Benutzer können die Rollen annehmen.

Informationen dazu, wie Sie unter Verwendung dieser beispielhaften JSON-Richtliniendokumente eine identitätsbasierte IAM-Richtlinie erstellen, finden Sie unter [Erstellen von IAM-Richtlinien](#) im IAM-Benutzerhandbuch.

Einzelheiten zu Aktionen und Ressourcentypen, die von Amazon Comprehend definiert werden, einschließlich des Formats der ARNs für die einzelnen Ressourcentypen, finden Sie unter [Aktionen, Ressourcen und Bedingungsschlüssel für Amazon Comprehend](#) in der Service-Autorisierungs-Referenz.

Themen

- [Bewährte Methoden für Richtlinien](#)
- [Verwenden der Amazon Comprehend-Konsole](#)

- [Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer](#)
- [Erforderliche Berechtigungen zum Ausführen von Dokumentanalyseaktionen](#)
- [Erforderliche Berechtigungen für die Verwendung der KMS-Verschlüsselung](#)
- [AWS Von verwaltete \(vordefinierte\) Richtlinien für Amazon Comprehend](#)
- [Rollenbasierte Berechtigungen, die für asynchrone Operationen erforderlich sind](#)
- [Berechtigungen zum Zulassen aller Amazon Comprehend-Aktionen](#)
- [Berechtigungen zum Zulassen von Aktionen zur Themenmodellierung](#)
- [Erforderliche Berechtigungen für einen benutzerdefinierten asynchronen Analyseauftrag](#)

## Bewährte Methoden für Richtlinien

Identitätsbasierte Richtlinien legen fest, ob jemand Amazon Comprehend-Ressourcen in Ihrem Konto erstellen, darauf zugreifen oder sie löschen kann. Dies kann zusätzliche Kosten für Ihr verursachen AWS-Konto. Befolgen Sie beim Erstellen oder Bearbeiten identitätsbasierter Richtlinien die folgenden Anleitungen und Empfehlungen:

- **Erste Schritte mit AWS-verwaltete Richtlinien und Umstellung auf Berechtigungen mit den geringsten Berechtigungen:** Um Ihren Benutzern und Workloads Berechtigungen zu gewähren, verwenden Sie die AWS-verwaltete Richtlinien die Berechtigungen für viele allgemeine Anwendungsfälle gewähren. Sie sind in Ihrem AWS-Konto verfügbar. Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie vom Kunden verwaltete AWS-Richtlinien definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind. Weitere Informationen finden Sie unter [AWS-verwaltete Richtlinien](#) oder [AWS-verwaltete Richtlinien für Auftragsfunktionen](#) im IAM-Benutzerhandbuch.
- **Anwendung von Berechtigungen mit den geringsten Rechten:** Wenn Sie mit IAM-Richtlinien Berechtigungen festlegen, gewähren Sie nur die Berechtigungen, die für die Durchführung einer Aufgabe erforderlich sind. Sie tun dies, indem Sie die Aktionen definieren, die für bestimmte Ressourcen unter bestimmten Bedingungen durchgeführt werden können, auch bekannt als die geringsten Berechtigungen. Weitere Informationen zur Verwendung von IAM zum Anwenden von Berechtigungen finden Sie unter [Richtlinien und Berechtigungen in IAM](#) im IAM-Benutzerhandbuch.
- **Verwenden von Bedingungen in IAM-Richtlinien zur weiteren Einschränkung des Zugriffs:** Sie können Ihren Richtlinien eine Bedingung hinzufügen, um den Zugriff auf Aktionen und Ressourcen zu beschränken. Sie können beispielsweise eine Richtlinienbedingung schreiben, um festzulegen,

dass alle Anforderungen mithilfe von SSL gesendet werden müssen. Sie können auch Bedingungen verwenden, um Zugriff auf Service-Aktionen zu gewähren, wenn diese durch ein bestimmtes AWS-Service, wie beispielsweise AWS CloudFormation, verwendet werden. Weitere Informationen finden Sie unter [IAM-JSON-Richtlinienelemente: Bedingung](#) im IAM-Benutzerhandbuch.

- Verwenden von IAM Access Analyzer zur Validierung Ihrer IAM-Richtlinien, um sichere und funktionale Berechtigungen zu gewährleisten: IAM Access Analyzer validiert neue und vorhandene Richtlinien, damit die Richtlinien der IAM-Richtliniensprache (JSON) und den bewährten IAM-Methoden entsprechen. IAM Access Analyzer stellt mehr als 100 Richtlinienprüfungen und umsetzbare Empfehlungen zur Verfügung, damit Sie sichere und funktionale Richtlinien erstellen können. Weitere Informationen finden Sie unter [Richtlinienvvalidierung zum IAM Access Analyzer](#) im IAM-Benutzerhandbuch.
- Bedarf einer Multi-Faktor-Authentifizierung (MFA): Wenn Sie ein Szenario haben, das IAM-Benutzer oder Root-Benutzer in Ihrem AWS-Konto erfordert, aktivieren Sie MFA für zusätzliche Sicherheit. Um MFA beim Aufrufen von API-Vorgängen anzufordern, fügen Sie Ihren Richtlinien MFA-Bedingungen hinzu. Weitere Informationen finden Sie unter [Konfigurieren eines MFA-geschützten API-Zugriffs](#) im IAM-Benutzerhandbuch.

Weitere Informationen zu bewährten Methoden in IAM finden Sie unter [Bewährte Methoden für die Sicherheit in IAM](#) im IAM-Benutzerhandbuch.

## Verwenden der Amazon Comprehend-Konsole

Um auf die Amazon Comprehend-Konsole zugreifen zu können, müssen Sie über einen Mindestsatz von Berechtigungen verfügen. Diese Berechtigungen müssen es Ihnen ermöglichen, Details zu den Amazon Comprehend-Ressourcen in Ihrem aufzulisten und anzuzeigen AWS-Konto. Wenn Sie eine identitätsbasierte Richtlinie erstellen, die strenger ist als die mindestens erforderlichen Berechtigungen, funktioniert die Konsole nicht wie vorgesehen für Entitäten (Benutzer oder Rollen) mit dieser Richtlinie.

Für Benutzer, die nur Aufrufe an die AWS CLI oder AWS-API durchführen, müssen Sie keine Mindestberechtigungen in der Konsole erteilen. Stattdessen sollten Sie nur Zugriff auf die Aktionen zulassen, die der API-Operation entsprechen, die die Benutzer ausführen möchten.

Für Mindestberechtigungen für die Amazon Comprehend-Konsole können Sie die *ComprehendReadOnly* AWS verwaltete Richtlinie an die Entitäten anfügen. Weitere Informationen finden Sie unter [Hinzufügen von Berechtigungen zu einem Benutzer](#) im IAM-Benutzerhandbuch.

Um die Amazon Comprehend-Konsole verwenden zu können, benötigen Sie auch Berechtigungen für die in der folgenden Richtlinie aufgeführten Aktionen:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "iam:ListRoles",
        "iam:GetRole",
        "s3:ListAllMyBuckets",
        "s3:ListBucket",
        "s3:GetBucketLocation"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

Die Amazon Comprehend-Konsole benötigt diese zusätzlichen Berechtigungen aus folgenden Gründen:

- iam -Berechtigungen zum Auflisten der verfügbaren IAM-Rollen für Ihr Konto.
- s3 -Berechtigungen für den Zugriff auf die Amazon S3-Buckets und -Objekte, die die Daten für die Themenmodellierung enthalten.

Wenn Sie einen asynchronen Batch-Auftrag oder einen Themenmodellierungsauftrag mit der Konsole erstellen, haben Sie die Möglichkeit, die Konsole eine IAM-Rolle für Ihren Auftrag erstellen zu lassen. Um eine IAM-Rolle zu erstellen, müssen Benutzer die folgenden zusätzlichen Berechtigungen erhalten, um IAM-Rollen und -Richtlinien zu erstellen und Richtlinien an Rollen anzufügen:

```
{
  "Version": "2012-10-17",
  "Statement":
  [
    {
      "Action":
      [
        "iam:CreateRole",
        "iam:CreatePolicy",
```

```
        "iam:AttachRolePolicy"
      ],
      "Effect": "Allow",
      "Resource": "*"
    },
    {
      "Action":
      [
        "iam:PassRole"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:iam::*:role/*Comprehend*"
    }
  ]
}
```

Die Amazon Comprehend-Konsole benötigt diese zusätzlichen Berechtigungen aus folgenden Gründen:

- `iam`-Berechtigungen zum Erstellen von Rollen und Richtlinien und zum Anfügen von Rollen und Richtlinien. Die `iam:PassRole` Aktion ermöglicht es der Konsole, die Rolle an Amazon Comprehend zu übergeben.

## Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer

In diesem Beispiel wird gezeigt, wie Sie eine Richtlinie erstellen, die IAM-Benutzern die Berechtigung zum Anzeigen der eingebundenen Richtlinien und verwalteten Richtlinien gewährt, die ihrer Benutzeridentität angefügt sind. Diese Richtlinie enthält Berechtigungen für die Ausführung dieser Aktion auf der Konsole oder für die programmgesteuerte Ausführung über die AWS CLI oder die AWS-API.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
        "iam:GetUserPolicy",
        "iam:ListGroupsWithUser",
        "iam:ListAttachedUserPolicies",

```

```

        "iam:ListUserPolicies",
        "iam:GetUser"
    ],
    "Resource": ["arn:aws:iam::*:user/${aws:username}"]
},
{
    "Sid": "NavigateInConsole",
    "Effect": "Allow",
    "Action": [
        "iam:GetGroupPolicy",
        "iam:GetPolicyVersion",
        "iam:GetPolicy",
        "iam:ListAttachedGroupPolicies",
        "iam:ListGroupPolicies",
        "iam:ListPolicyVersions",
        "iam:ListPolicies",
        "iam:ListUsers"
    ],
    "Resource": "*"
}
]
}

```

## Erforderliche Berechtigungen zum Ausführen von Dokumentanalyseaktionen

Die folgende Beispielrichtlinie gewährt Berechtigungen zur Verwendung der Amazon Comprehend-Dokumentanalyseaktionen:

```

{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "AllowDetectActions",
    "Effect": "Allow",
    "Action": [
      "comprehend:DetectEntities",
      "comprehend:DetectKeyPhrases",
      "comprehend:DetectDominantLanguage",
      "comprehend:DetectSentiment",
      "comprehend:DetectTargetedSentiment",
      "comprehend:DetectSyntax",
      "textract:DetectDocumentText",
      "textract:AnalyzeDocument"
    ],
  ],
}

```



```
    "Resource": "*"
  }
]
```

Die Richtlinie enthält eine Anweisung, die die Berechtigung zur Verwendung der DetectSyntax Aktionen DetectEntities, DetectKeyPhrases, DetectTargetedSentiment, DetectDominantLanguageDetectSentiment, und gewährt. Die Richtlinienanweisung gewährt auch Berechtigungen zur Verwendung von zwei Amazon Textract-API-Methoden. Amazon Comprehend ruft diese Methoden auf, um Text aus Bilddateien und gescannten PDF-Dokumenten zu extrahieren. Sie können diese Berechtigungen für Benutzer entfernen, die niemals benutzerdefinierte Inferenzen für diese Arten von Eingabedateien ausführen.

Ein Benutzer mit dieser Richtlinie wäre nicht in der Lage, Batch-Aktionen oder asynchrone Aktionen in Ihrem Konto durchzuführen.

Die Richtlinie gibt nicht das Principal-Element an, da in einer identitätsbasierten Richtlinie nicht der Prinzipal angegeben wird, der die Berechtigung erhält. Wenn Sie einem Benutzer eine Richtlinie anfügen, ist der Benutzer automatisch der Prinzipal. Wird die Berechtigungsrichtlinie einer IAM-Rolle angefügt, erhält der in der Vertrauensrichtlinie der Rolle angegebene Prinzipal die Berechtigungen.

Eine Tabelle mit allen Amazon Comprehend-API-Aktionen und den Ressourcen, für die sie gelten, finden Sie unter [Aktionen, Ressourcen und Bedingungsschlüssel für Amazon Comprehend](#) in der Service-Autorisierungs-Referenz.

## Erforderliche Berechtigungen für die Verwendung der KMS-Verschlüsselung

Um Amazon Key Management Service (KMS) für Daten und Auftragsverschlüsselung in einem asynchronen Auftrag vollständig zu verwenden, müssen Sie Berechtigungen für die in der folgenden Richtlinie aufgeführten Aktionen erteilen:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "kms:CreateGrant"
      ],
      "Effect": "Allow",
```

```
    "Resource": "*"
  },
  {
    "Action": [
      "kms:Decrypt",
      "kms:GenerateDatakey"
    ],
    "Effect": "Allow",
    "Resource": "*",
    "Condition": {
      "StringEquals": {
        "kms:ViaService": [
          "s3.region.amazonaws.com"
        ]
      }
    }
  }
]
```

Wenn Sie einen asynchronen Auftrag mit Amazon Comprehend erstellen, verwenden Sie Eingabedaten, die auf Amazon S3 gespeichert sind. Mit S3 haben Sie die Möglichkeit, Ihre gespeicherten Daten zu verschlüsseln, die mit S3 und nicht mit Amazon Comprehend verschlüsselt sind. Wir können diese verschlüsselten Eingabedaten entschlüsseln und lesen, wenn Sie die `kms:Decrypt` Berechtigung für den Schlüssel erteilen, mit dem die ursprünglichen Eingabedaten mit der Datenzugriffsrolle verschlüsselt wurden, die vom Amazon Comprehend-Auftrag verwendet wird.

Sie haben auch die Möglichkeit, vom Kunden verwaltete KMS-Schlüssel (CMK) zum Verschlüsseln der Ausgabeergebnisse auf S3 sowie des Speichervolumens zu verwenden, das während der Auftragsverarbeitung verwendet wird. Wenn Sie dies tun, können Sie denselben KMS-Schlüssel für beide Verschlüsselungstypen verwenden, dies ist jedoch nicht erforderlich. Beim Erstellen des Auftrags sind separate Felder verfügbar, um die Schlüssel für die Ausgabeverschlüsselung und Volume-Verschlüsselung anzugeben, und Sie können sogar einen KMS-Schlüssel aus einem anderen Konto verwenden.

Bei Verwendung der KMS-Verschlüsselung ist die `-kms:CreateGrant` Berechtigung für die Volume-Verschlüsselung und die `-kms:GenerateDataKey` Berechtigung für die Ausgabedatenverschlüsselung erforderlich. Zum Lesen verschlüsselter Eingaben (wie wenn die Eingabedaten bereits mit Amazon S3 verschlüsselt sind) ist eine `-kms:Decrypt` Berechtigung erforderlich. Die IAM-Rolle muss diese Berechtigungen nach Bedarf erteilen. Wenn der Schlüssel jedoch von einem anderen Konto stammt, als derzeit verwendet wird, muss die KMS-

Schlüsselrichtlinie für diesen kms-Schlüssel diese Berechtigungen auch der Datenzugriffsrolle für den Auftrag erteilen.

## AWS Von verwaltete (vordefinierte) Richtlinien für Amazon Comprehend

Durch die Bereitstellung von eigenständigen IAM-Richtlinien, die von AWS erstellt und verwaltet werden, deckt AWS viele häufige Anwendungsfälle ab. Diese von AWS verwalteten Richtlinien erteilen die erforderlichen Berechtigungen für viele häufige Anwendungsfälle, sodass Sie nicht mühsam ermitteln müssen, welche Berechtigungen erforderlich sind. Weitere Informationen finden Sie unter [Von AWS verwaltete Richtlinien](#) im IAM-Benutzerhandbuch.

Die folgenden AWS verwalteten Richtlinien, die Sie Benutzern in Ihrem Konto anfügen können, gelten speziell für Amazon Comprehend :

- **ComprehendFullAccess** – Gewährt vollen Zugriff auf Amazon Comprehend-Ressourcen, einschließlich der Ausführung von Themenmodellierungsaufträgen. Enthält die Berechtigung zum Auflisten und Abrufen von IAM-Rollen.
- **ComprehendReadOnly** – Gewährt die Berechtigung zum Ausführen aller Amazon Comprehend-Aktionen außer `StartDominantLanguageDetectionJob`, `StartEntitiesDetectionJob`, `StartKeyPhrasesDetectionJob`, `StartSentimentDetectionJob` `StartTargetedSentimentDetectionJob`, und `StartTopicsDetectionJob`.

Sie müssen die folgende zusätzliche Richtlinie auf alle Benutzer anwenden, die Amazon Comprehend verwenden:

```
{
  "Version": "2012-10-17",
  "Statement":
  [
    {
      "Action":
      [
        "iam:PassRole"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:iam::*:role/*Comprehend*"
    }
  ]
}
```

```
}
```

Sie können die verwalteten Berechtigungsrichtlinien überprüfen, indem Sie sich bei der IAM-Konsole anmelden und dort nach bestimmten Richtlinien suchen.

Diese Richtlinien funktionieren, wenn Sie AWS -SDKs oder die -AWSCLI verwenden.

Sie können auch Ihre eigenen benutzerdefinierten IAM-Richtlinien erstellen, um Berechtigungen für Amazon Comprehend-Aktionen und -Ressourcen zu gewähren. Sie können diese benutzerdefinierten Richtlinien den Benutzern, Gruppen oder Rollen anfügen, die diese Berechtigungen benötigen.

## Rollenbasierte Berechtigungen, die für asynchrone Operationen erforderlich sind

Um die asynchronen Operationen von Amazon Comprehend verwenden zu können, müssen Sie Amazon Comprehend Zugriff auf den Amazon S3-Bucket gewähren, der Ihre Dokumentsammlung enthält. Dazu erstellen Sie eine Datenzugriffsrolle in Ihrem Konto mit einer Vertrauensrichtlinie, um dem Amazon Comprehend-Service-Prinzipal zu vertrauen. Weitere Informationen zum Erstellen einer Rolle finden Sie unter [Erstellen einer Rolle zum Delegieren von Berechtigungen an einen - AWS-Service](#) im AWS Benutzerhandbuch für Identity and Access Management.

Im Folgenden sehen Sie ein Beispiel für eine Vertrauensrichtlinie für die Rolle, die Sie erstellen. Um bei der [Prävention von verwirrtem Stellvertreter](#) zu helfen, schränken Sie den Umfang der Berechtigung ein, indem Sie einen oder mehrere globale Bedingungskontextschlüssel verwenden. Legen Sie den Wert `aws:SourceAccount` auf Ihre Konto-ID fest. Wenn Sie die `ArnEquals` Bedingung verwenden, legen Sie den `aws:SourceArn` Wert auf den ARN des Auftrags fest. Verwenden Sie einen Platzhalter für die Auftragsnummer im ARN, da Amazon Comprehend diese Nummer im Rahmen der Auftragserstellung generiert.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "comprehend.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "111122223333"
        }
      }
    }
  ]
}
```

```
        "ArnEquals": {
            "aws:SourceArn": "arn:aws:comprehend:us-west-2:111122223333:pii-entities-
detection-job/*"
        }
    }
}
]
```

Nachdem Sie die Rolle erstellt haben, erstellen Sie eine Zugriffsrichtlinie für diese Rolle. Dadurch sollten die Amazon S3- GetObject und -ListBucketBerechtigungen für den Amazon S3-Bucket, der Ihre Eingabedaten enthält, und die Amazon S3-PutObjectBerechtigung für Ihren Amazon S3-Ausgabedaten-Bucket erteilt werden.

## Berechtigungen zum Zulassen aller Amazon Comprehend-Aktionen

Nachdem Sie sich bei registriert habenAWS, erstellen Sie einen Administratorbenutzer zur Verwaltung Ihres -Kontos, einschließlich der Erstellung von Benutzern und der Verwaltung ihrer Berechtigungen.

Sie können einen Benutzer erstellen, der über Berechtigungen für alle Amazon Comprehend-Aktionen verfügt (d. h. für diesen Benutzer als servicespezifischer Administrator), um mit Amazon Comprehend zu arbeiten. Diesem Benutzer können Sie die folgende Berechtigungsrichtlinie zuweisen:

```
{
  "Version": "2012-10-17",
  "Statement":
  [
    {
      "Sid": "AllowAllComprehendActions",
      "Effect": "Allow",
      "Action":
      [
        "comprehend:*",
        "iam:ListRoles",
        "iam:GetRole",
        "s3:ListAllMyBuckets",
        "s3:ListBucket",
        "s3:GetBucketLocation",
        "iam:CreateRole",
        "iam:CreatePolicy",
```

```
        "iam:AttachRolePolicy",
        "kms:CreateGrant",
        "kms:Decrypt",
        "kms:GenerateDatakey"
    ],
    "Resource": "*"
},
{
    "Action":
    [
        "iam:PassRole"
    ],
    "Effect": "Allow",
    "Resource": "arn:aws:iam::*:role/*Comprehend*"
}
]
```

Diese Berechtigungen können in Bezug auf die Verschlüsselung wie folgt geändert werden:

- Damit Amazon Comprehend Dokumente analysieren kann, die in einem verschlüsselten S3-Bucket gespeichert sind, muss die IAM-Rolle über die `-kms:Decrypt` Berechtigung verfügen.
- Damit Amazon Comprehend Dokumente verschlüsseln kann, die auf einem Speicher-Volume gespeichert sind, das der Rechen-Instance angefügt ist, die den Analyseauftrag verarbeitet, muss die IAM-Rolle über die `-kms:CreateGrant` Berechtigung verfügen.
- Damit Amazon Comprehend die Ausgabeergebnisse in seinem S3-Bucket verschlüsseln kann, muss die IAM-Rolle über die `-kms:GenerateDataKey` Berechtigung verfügen.

## Berechtigungen zum Zulassen von Aktionen zur Themenmodellierung

Die folgende Berechtigungsrichtlinie gewährt Benutzern Berechtigungen zum Ausführen der Amazon Comprehend-Themenmodellierungsvorgänge.

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Sid": "AllowTopicModelingActions",
        "Effect": "Allow",
        "Action": [
            "comprehend:DescribeTopicsDetectionJob",
            "comprehend:ListTopicsDetectionJobs",
```

```

        "comprehend:StartTopicsDetectionJob",
    ],
    "Resource": "*"
  ]
}
]
}

```

## Erforderliche Berechtigungen für einen benutzerdefinierten asynchronen Analyseauftrag

### Important

Wenn Sie über eine IAM-Richtlinie verfügen, die den Modellzugriff einschränkt, können Sie keinen Inferenzauftrag mit einem benutzerdefinierten Modell abschließen. Ihre IAM-Richtlinie sollte aktualisiert werden, um eine Platzhalterressource für einen benutzerdefinierten asynchronen Analyseauftrag zu haben.

Wenn Sie die [StartEntitiesDetectionJob](#) APIs [StartDocumentClassificationJob](#) und verwenden, müssen Sie Ihre IAM-Richtlinie aktualisieren, es sei denn, Sie verwenden derzeit Platzhalter als Ressourcen. Wenn Sie ein [StartEntitiesDetectionJob](#) vortrainiertes Modell verwenden, hat dies keine Auswirkungen auf Sie und Sie müssen keine Änderungen vornehmen.

Die folgende Beispielrichtlinie enthält eine veraltete Referenz.

```

{
  "Action": [
    "comprehend:StartDocumentClassificationJob",
    "comprehend:StartEntitiesDetectionJob",
  ],
  "Resource": [
    "arn:aws:comprehend:us-east-1:123456789012:document-classifier/myClassifier",
    "arn:aws:comprehend:us-east-1:123456789012:entity-recognizer/myRecognizer"
  ],
  "Effect": "Allow"
}

```

Dies ist die aktualisierte Richtlinie, die Sie verwenden müssen, um [StartDocumentClassificationJob](#) und erfolgreich auszuführen [StartEntitiesDetectionJob](#).

```
{
  "Action": [
    "comprehend:StartDocumentClassificationJob",
    "comprehend:StartEntitiesDetectionJob",
  ],
  "Resource": [
    "arn:aws:comprehend:us-east-1:123456789012:document-classifier/myClassifier",
    "arn:aws:comprehend:us-east-1:123456789012:document-classification-job/*",
    "arn:aws:comprehend:us-east-1:123456789012:entity-recognizer/myRecognizer",
    "arn:aws:comprehend:us-east-1:123456789012:entities-detection-job/*"
  ],
  "Effect": "Allow"
}
```

## AWS Von verwaltete Richtlinien für Amazon Comprehend

Um Benutzern, Gruppen und Rollen Berechtigungen hinzuzufügen, ist es einfacher, von AWS verwaltete Richtlinien zu verwenden, als selbst Richtlinien zu schreiben. Es erfordert Zeit und Fachwissen, um [von Kunden verwaltete IAM-Richtlinien zu erstellen](#), die Ihrem Team nur die benötigten Berechtigungen bieten. Um schnell loszulegen, können Sie unsere von AWS verwalteten Richtlinien verwenden. Diese Richtlinien decken allgemeine Anwendungsfälle ab und sind in Ihrem AWS-Konto verfügbar. Weitere Informationen zu verwalteten AWS-Richtlinien finden Sie unter [Verwaltete AWS-Richtlinien](#) im IAM-Leitfaden.

AWS-Services pflegen und Aktualisieren von verwalteten AWS-Richtlinien. Die Berechtigungen in von AWS verwalteten Richtlinien können nicht geändert werden. Services fügen einer von AWS verwalteten Richtlinien gelegentlich zusätzliche Berechtigungen hinzu, um neue Features zu unterstützen. Diese Art von Update betrifft alle Identitäten (Benutzer, Gruppen und Rollen), an welche die Richtlinie angehängt ist. Services aktualisieren eine von AWS verwaltete Richtlinie am ehesten, ein neues Feature gestartet wird oder neue Vorgänge verfügbar werden. Services entfernen keine Berechtigungen aus einer von AWS verwalteten Richtlinie, so dass Richtlinien-Aktualisierungen Ihre vorhandenen Berechtigungen nicht beeinträchtigen.

Darüber hinaus unterstützt AWS verwaltete Richtlinien für Auftragsfunktionen, die mehrere Services umfassen. Die ReadOnlyAccess AWS verwaltete Richtlinie bietet beispielsweise schreibgeschützten Zugriff auf alle AWS Services und -Ressourcen. Wenn ein Service ein neues Feature startet, fügt AWS schreibgeschützte Berechtigungen für neue Vorgänge und Ressourcen hinzu. Eine Liste und Beschreibungen der Richtlinien für Auftragsfunktionen finden Sie in [Verwaltete AWS-Richtlinien für Auftragsfunktionen](#) im IAM-Leitfaden.



## AWS verwaltete Richtlinie: ComprehendFullAccess

Diese Richtlinie gewährt vollen Zugriff auf Amazon Comprehend-Ressourcen, einschließlich der Ausführung von Themenmodellierungsaufträgen. Diese Richtlinie gewährt auch Listen- und Abrufberechtigungen für Amazon S3-Buckets und IAM-Rollen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "comprehend:*",
        "iam:GetRole",
        "iam:ListRoles",
        "s3:GetBucketLocation",
        "s3:ListAllMyBuckets",
        "s3:ListBucket",
      ],
      "Resource": "*"
    }
  ]
}
```

## AWS verwaltete Richtlinie: ComprehendReadOnly

Diese Richtlinie gewährt Leseberechtigungen zum Ausführen aller Amazon Comprehend-Aktionen mit Ausnahme der folgenden:

- StartDominantLanguageDetectionJob
- StartEntitiesDetectionJob
- StartKeyPhrasesDetectionJob
- StartSentimentDetectionJob
- StartTargetedSentimentDetectionJob
- StartTopicsDetectionJob

```
{
```

```
"Version": "2012-10-17",
"Statement": [
  {
    "Action": [
      "comprehend:BatchDetectDominantLanguage",
      "comprehend:BatchDetectEntities",
      "comprehend:BatchDetectKeyPhrases",
      "comprehend:BatchDetectSentiment",
      "comprehend:BatchDetectSyntax",
      "comprehend:ClassifyDocument",
      "comprehend:ContainsPiiEntities",
      "comprehend:DescribeDocumentClassificationJob",
      "comprehend:DescribeDocumentClassifier",
      "comprehend:DescribeDominantLanguageDetectionJob",
      "comprehend:DescribeEndpoint",
      "comprehend:DescribeEntitiesDetectionJob",
      "comprehend:DescribeEntityRecognizer",
      "comprehend:DescribeKeyPhrasesDetectionJob",
      "comprehend:DescribePiiEntitiesDetectionJob",
      "comprehend:DescribeResourcePolicy",
      "comprehend:DescribeSentimentDetectionJob",
      "comprehend:DescribeTargetedSentimentDetectionJob",
      "comprehend:DescribeTopicsDetectionJob",
      "comprehend:DetectDominantLanguage",
      "comprehend:DetectEntities",
      "comprehend:DetectKeyPhrases",
      "comprehend:DetectPiiEntities",
      "comprehend:DetectSentiment",
      "comprehend:DetectSyntax",
      "comprehend:ListDocumentClassificationJobs",
      "comprehend:ListDocumentClassifiers",
      "comprehend:ListDocumentClassifierSummaries",
      "comprehend:ListDominantLanguageDetectionJobs",
      "comprehend:ListEndpoints",
      "comprehend:ListEntitiesDetectionJobs",
      "comprehend:ListEntityRecognizers",
      "comprehend:ListEntityRecognizerSummaries",
      "comprehend:ListKeyPhrasesDetectionJobs",
      "comprehend:ListPiiEntitiesDetectionJobs",
      "comprehend:ListSentimentDetectionJobs",
      "comprehend:ListTargetedSentimentDetectionJobs",
      "comprehend:ListTagsForResource",
      "comprehend:ListTopicsDetectionJobs"
    ]
  }
],
```

```

        "Effect": "Allow",
        "Resource": "*"
    }
]
}

```

## Amazon Comprehend-Updates für -AWSverwaltete Richtlinien

Anzeigen von Details zu Aktualisierungen für -AWSverwaltete Richtlinien für Amazon Comprehend, seit dieser Service mit der Verfolgung dieser Änderungen begonnen hat. Um automatische Warnungen über Änderungen an dieser Seite zu erhalten, abonnieren Sie den RSS-Feed auf der Seite Amazon Comprehend [Document History](#).

Änderung	Beschreibung	Datum
<a href="#">ComprehendReadOnly</a> – Aktualisierung auf eine bestehende Richtlinie	Amazon Comprehend erlaubt jetzt die <code>comprehend:ListTargetedSentimentDetectionJobs</code> Aktionen <code>comprehend:DescribeTargetedSentimentDetectionJob</code> und in der <code>ComprehendReadOnly</code> Richtlinie	30. März 2022
<a href="#">ComprehendReadOnly</a> – Aktualisierung auf eine bestehende Richtlinie	Amazon Comprehend erlaubt jetzt die <code>comprehend:DescribeResourcePolicy</code> Aktion in der <code>ComprehendReadOnly</code> Richtlinie	2. Februar 2022
<a href="#">ComprehendReadOnly</a> – Aktualisierung auf eine bestehende Richtlinie	Amazon Comprehend erlaubt jetzt die <code>ListEntityRecognizerSummaries</code> Aktionen <code>ListDocumentClassifierSumma</code>	21. September 2021

Änderung	Beschreibung	Datum
	ries und in der ComprehendReadOnly Richtlinie	
<a href="#">ComprehendReadOnly</a> – Aktualisierung auf eine bestehende Richtlinie	Amazon Comprehend erlaubt jetzt die ContainsPIIEntities Aktion in der ComprehendReadOnly Richtlinie	26. März 2021
Amazon Comprehend hat mit der Verfolgung von Änderungen begonnen	Amazon Comprehend hat mit der Verfolgung von Änderungen für seine von AWS verwalteten Richtlinien begonnen.	1. März 2021

## Fehlerbehebung bei Amazon Comprehend Identity and Access

Verwenden Sie die folgenden Informationen, um häufig auftretende Probleme zu diagnostizieren und zu beheben, die bei der Arbeit mit Amazon Comprehend und IAM auftreten können.

### Themen

- [Ich bin nicht berechtigt, eine Aktion in Amazon Comprehend durchzuführen](#)
- [Ich bin nicht berechtigt, iam durchzuführen: PassRole](#)
- [Ich möchte Personen außerhalb von mir den Zugriff auf meine AWS-Konto Amazon Comprehend Comprehend-Ressourcen ermöglichen](#)

### Ich bin nicht berechtigt, eine Aktion in Amazon Comprehend durchzuführen

Wenn Sie eine Fehlermeldung erhalten, dass Sie nicht zur Durchführung einer Aktion berechtigt sind, müssen Ihre Richtlinien aktualisiert werden, damit Sie die Aktion durchführen können.

Der folgende Beispielfehler tritt auf, wenn der mateojackson IAM-Benutzer versucht, die Konsole zum Anzeigen von Details zu einer fiktiven *my-example-widget*-Ressource zu verwenden, jedoch nicht über comprehend: *GetWidget*-Berechtigungen verfügt.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
comprehend:GetWidget on resource: my-example-widget
```

In diesem Fall muss die Mateo-Richtlinie aktualisiert werden, damit er mit der `comprehend:GetWidget`-Aktion auf die `my-example-widget`-Ressource zugreifen kann.

Wenn Sie Hilfe benötigen, wenden Sie sich an Ihren AWS Administrator. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

## Ich bin nicht berechtigt, iam durchzuführen: PassRole

Wenn Sie eine Fehlermeldung erhalten, dass Sie nicht berechtigt sind, die `iam:PassRole` Aktion durchzuführen, müssen Ihre Richtlinien aktualisiert werden, damit Sie eine Rolle an Amazon Comprehend übergeben können.

Einige AWS-Services ermöglichen es Ihnen, eine bestehende Rolle an diesen Service zu übergeben, anstatt eine neue Servicerolle oder eine dienstbezogene Rolle zu erstellen. Hierzu benötigen Sie Berechtigungen für die Übergabe der Rolle an den Dienst.

Der folgende Beispielfehler tritt auf, wenn ein IAM-Benutzer mit dem Namen `marymajor` versucht, die Konsole zu verwenden, um eine Aktion in Amazon Comprehend auszuführen. Die Aktion erfordert jedoch, dass der Service über Berechtigungen verfügt, die durch eine Servicerolle gewährt werden. Mary besitzt keine Berechtigungen für die Übergabe der Rolle an den Dienst.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

In diesem Fall müssen die Richtlinien von Mary aktualisiert werden, um die Aktion `iam:PassRole` ausführen zu können.

Wenn Sie Hilfe benötigen, wenden Sie sich an Ihren Administrator. AWS Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

## Ich möchte Personen außerhalb von mir den Zugriff auf meine AWS-Konto Amazon Comprehend Comprehend-Ressourcen ermöglichen

Sie können eine Rolle erstellen, die Benutzer in anderen Konten oder Personen außerhalb Ihrer Organisation für den Zugriff auf Ihre Ressourcen verwenden können. Sie können festlegen, wem die Übernahme der Rolle anvertraut wird. Im Fall von Diensten, die ressourcenbasierte Richtlinien

oder Zugriffskontrolllisten (Access Control Lists, ACLs) verwenden, können Sie diese Richtlinien verwenden, um Personen Zugriff auf Ihre Ressourcen zu gewähren.

Weitere Informationen dazu finden Sie hier:

- Informationen darüber, ob Amazon Comprehend diese Funktionen unterstützt, finden Sie unter [So funktioniert Amazon Comprehend mit IAM](#)
- Informationen dazu, wie Sie Zugriff auf Ihre Ressourcen gewähren können, AWS-Konten die Ihnen gehören, finden Sie im IAM-Benutzerhandbuch unter [Gewähren des Zugriffs auf einen IAM-Benutzer in einem anderen AWS-Konto , den Sie besitzen](#).
- Informationen dazu, wie Sie Dritten Zugriff auf Ihre Ressourcen gewähren können AWS-Konten, finden Sie [AWS-Konten im IAM-Benutzerhandbuch unter Gewähren des Zugriffs für Dritte](#).
- Informationen dazu, wie Sie über einen Identitätsverbund Zugriff gewähren, finden Sie unter [Gewähren von Zugriff für extern authentifizierte Benutzer \(Identitätsverbund\)](#) im IAM-Benutzerhandbuch.
- Informationen zum Unterschied zwischen der Verwendung von Rollen und ressourcenbasierten Richtlinien für den kontoübergreifenden Zugriff finden Sie im IAM-Benutzerhandbuch unter [Kontenübergreifender Ressourcenzugriff in IAM](#).

## Protokollieren von Amazon Comprehend-API-Aufrufen mit AWS CloudTrail

Amazon Comprehend ist in integriert, einem ServiceAWS CloudTrail, der eine Aufzeichnung der von einem Benutzer, einer Rolle oder einem -AWSService in Amazon Comprehend durchgeführten Aktionen bietet. CloudTrail captures API-Aufrufe für Amazon Comprehend als Ereignisse. Zu den erfassten Aufrufen gehören Aufrufe von der Amazon Comprehend-Konsole und Codeaufrufe der Amazon Comprehend-API-Operationen. Wenn Sie einen Trail erstellen, können Sie die kontinuierliche Bereitstellung von CloudTrail Ereignissen an einen Amazon S3-Bucket aktivieren, einschließlich Ereignissen für Amazon Comprehend . Wenn Sie keinen Trail konfigurieren, können Sie trotzdem die neuesten Ereignisse in der CloudTrail Konsole unter Ereignisverlauf anzeigen. Anhand der von CloudTrail gesammelten Informationen können Sie die an Amazon Comprehend gestellte Anfrage, die IP-Adresse, von der die Anfrage gestellt wurde, den Initiator der Anfrage, den Zeitpunkt der Anfrage und zusätzliche Details bestimmen.

Weitere Informationen zu CloudTrail, einschließlich Konfiguration und Aktivierung, finden Sie im [AWS CloudTrail -Benutzerhandbuch](#).

## Amazon Comprehend-Informationen in CloudTrail

CloudTrail wird beim Erstellen des Kontos AWS-Konto auf Ihrem aktiviert. Wenn die unterstützte Ereignisaktivität in Amazon Comprehend auftritt, wird diese Aktivität in einem - CloudTrail Ereignis zusammen mit anderen -AWSServiceereignissen im Ereignisverlauf aufgezeichnet. Sie können in Ihrem AWS-Konto die neusten Ereignisse anzeigen, suchen und herunterladen. Weitere Informationen finden Sie unter [Anzeigen von Ereignissen mit dem CloudTrail Ereignisverlauf](#) .

Erstellen Sie für eine fortlaufende Aufzeichnung der Ereignisse in Ihrem AWS-Konto, einschließlich Ereignissen für Amazon Comprehend, einen Trail. Ein Trail ermöglicht CloudTrail die Bereitstellung von Protokolldateien an einen Amazon S3-Bucket. Wenn Sie einen Trail in der Konsole anlegen, gilt dieser für alle AWS-Regionen. Der Trail protokolliert Ereignisse aus allen Regionen in der AWS-Partition und stellt die Protokolldateien in dem von Ihnen angegebenen Amazon-S3-Bucket bereit. Darüber hinaus können Sie andere -AWSServices konfigurieren, um die in den CloudTrail Protokollen erfassten Ereignisdaten weiter zu analysieren und entsprechend zu agieren. Weitere Informationen finden Sie hier:

- [Übersicht zum Erstellen eines Trails](#)
- [CloudTrail Von unterstützte Services und Integrationen](#)
- [Konfigurieren von Amazon SNS-Benachrichtigungen für CloudTrail](#)
- [Empfangen von CloudTrail Protokolldateien aus mehreren Regionen](#) und [Empfangen von CloudTrail Protokolldateien aus mehreren Konten](#)

Amazon Comprehend unterstützt die Protokollierung der folgenden Aktionen als Ereignisse in CloudTrail Protokolldateien:

- [BatchDetectDominantLanguage](#)
- [BatchDetectEntities](#)
- [BatchDetectKeyPhrases](#)
- [BatchDetectSentiment](#)
- [BatchDetectSyntax](#)
- [ClassifyDocument](#)
- [CreateDocumentClassifier](#)
- [CreateEndpoint](#)
- [CreateEntityRecognizer](#)

- [DeleteDocumentClassifier](#)
- [DeleteEndpoint](#)
- [DeleteEntityRecognizer](#)
- [DescribeDocumentClassificationJob](#)
- [DescribeDocumentClassifier](#)
- [DescribeDominantLanguageDetectionJob](#)
- [DescribeEndpoint](#)
- [DescribeEntitiesDetectionJob](#)
- [DescribeEntityRecognizer](#)
- [DescribeKeyPhrasesDetectionJob](#)
- [DescribePiiEntitiesDetectionJob](#)
- [DescribeSentimentDetectionJob](#)
- [DescribeTargetedSentimentDetectionJob](#)
- [DescribeTopicsDetectionJob](#)
- [DetectDominantLanguage](#)
- [DetectEntities](#)
- [DetectKeyPhrases](#)
- [DetectPiiEntities](#)
- [DetectSentiment](#)
- [DetectSyntax](#)
- [ListDocumentClassificationJobs](#)
- [ListDocumentClassifiers](#)
- [ListDominantLanguageDetectionJobs](#)
- [ListEndpoints](#)
- [ListEntitiesDetectionJobs](#)
- [ListEntityRecognizers](#)
- [ListKeyPhrasesDetectionJobs](#)
- [ListPiiEntitiesDetectionJobs](#)
- [ListSentimentDetectionJobs](#)



- [ListTargetedSentimentDetectionJobs](#)
- [ListTagsForResource](#)
- [ListTopicsDetectionJobs](#)
- [StartDocumentClassificationJob](#)
- [StartDominantLanguageDetectionJob](#)
- [StartEntitiesDetectionJob](#)
- [StartKeyPhrasesDetectionJob](#)
- [StartPiiEntitiesDetectionJob](#)
- [StartSentimentDetectionJob](#)
- [StartTargetedSentimentDetectionJob](#)
- [StartTopicsDetectionJob](#)
- [StopDominantLanguageDetectionJob](#)
- [StopEntitiesDetectionJob](#)
- [StopKeyPhrasesDetectionJob](#)
- [StopPiiEntitiesDetectionJob](#)
- [StopSentimentDetectionJob](#)
- [StopTargetedSentimentDetectionJob](#)
- [StopTrainingDocumentClassifier](#)
- [StopTrainingEntityRecognizer](#)
- [TagResource](#)
- [UntagResource](#)
- [UpdateEndpoint](#)

Jeder Ereignis- oder Protokolleintrag enthält Informationen zu dem Benutzer, der die Anforderung generiert hat. Die Identitätsinformationen unterstützen Sie bei der Ermittlung der folgenden Punkte:

- Ob die Anforderung mit den Anmeldeinformationen des Root-Benutzers ausgeführt wurde.
- Gibt an, ob die Anforderung mit temporären Sicherheitsanmeldeinformationen für eine Rolle oder einen Verbundbenutzer gesendet wurde.
- Ob die Anforderung aus einem anderen AWS-Service gesendet wurde

Weitere Informationen finden Sie unter [CloudTrail -Element userIdentity](#).

## Beispiel: Amazon Comprehend-Protokolldateieinträge

Ein Trail ist eine Konfiguration, die die Bereitstellung von Ereignissen als Protokolldateien an einen von Ihnen angegebenen Amazon S3-Bucket ermöglicht. CloudTrail Protokolldateien enthalten einen oder mehrere Protokolleinträge. Ein Ereignis stellt eine einzelne Anforderung aus einer beliebigen Quelle dar und enthält Informationen über die angeforderte Aktion, das Datum und die Uhrzeit der Aktion, Anforderungsparameter usw. CloudTrail Protokolldateien sind kein geordnetes Stacktrace der öffentlichen API-Aufrufe und erscheinen daher nicht in einer bestimmten Reihenfolge.

Das folgende Beispiel zeigt einen - CloudTrail Protokolleintrag, der die `ClassifyDocument` Aktion demonstriert.

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AROAI CFHPEXAMPLE",
    "arn": "arn:aws:iam::12345678910:user/myadmin2",
    "accountId": "12345678910",
    "accessKeyId": "ASIA3VZEXAMPLE",
    "sessionContext": {
      "sessionIssuer": {},
      "webIdFederationData": {},
      "attributes": {
        "creationDate": "2023-10-19T14:22:09Z",
        "mfaAuthenticated": "false"
      }
    }
  },
  "eventTime": "2023-10-19T17:31:20Z",
  "eventSource": "comprehend.amazonaws.com",
  "eventName": "ClassifyDocument",
  "awsRegion": "us-east-2",
  "sourceIPAddress": "3.21.185.237",
  "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:109.0)
Gecko/20100101 Firefox/115.0",
  "requestParameters": null,
  "responseElements": null,
  "requestID": "fd916e66-caac-46c9-a1fc-81a0ef33e61b",
  "eventID": "535ca22b-b3a3-4c13-b2c5-bf51ab082794",
```

```
"readOnly": false,
"resources": [
  {
    "accountId": "12345678910",
    "type": "AWS::Comprehend::DocumentClassifierEndpoint",
    "ARN": "arn:aws:comprehend:us-east-2:12345678910:document-classifier-
endpoint/endpointExample"
  }
],
"eventType": "AwsApiCall",
"recipientAccountId": "12345678910"
}
```

## Compliance-Validierung für Amazon Comprehend

Externe Prüfer bewerten im Rahmen verschiedener AWS -Compliance-Programme die Sicherheit und Compliance von Amazon Comprehend. Hierzu zählen unter anderem PCI, FedRAMP und HIPAA. Die Auditberichte von Drittanbietern lassen sich mit AWS Artifact herunterladen. Weitere Informationen finden Sie unter [Herunterladen von Berichten in AWS-Artifact](#).

Ihre Compliance-Verantwortung bei der Verwendung von Amazon Comprehend hängt von der Vertraulichkeit Ihrer Daten, den Compliance-Zielen Ihres Unternehmens und den geltenden Gesetzen und Vorschriften ab. AWS stellt die folgenden Ressourcen zur Unterstützung der Compliance bereit:

- [Kurzanleitungen für Sicherheit und Compliance](#) – In diesen Bereitstellungsleitfäden finden Sie wichtige Überlegungen zur Architektur sowie die einzelnen Schritte zur Bereitstellung von sicherheits- und Compliance-orientierten Basisumgebungen in AWS.
- [Whitepaper zur Erstellung einer Architektur mit HIPAA-konformer Sicherheit und Compliance](#) – In diesem Whitepaper wird beschrieben, wie Unternehmen mithilfe von AWS HIPAA-konforme Anwendungen erstellen können.
- [AWS-Compliance-Ressourcen](#) – Diese Arbeitsbücher und Leitfäden könnten für Ihre Branche und Ihren Standort interessant sein.
- [AWS Config](#) – Dieser AWS-Service bewertet, zu welchem Grad die Konfiguration Ihrer Ressourcen den internen Vorgehensweisen, Branchenrichtlinien und Vorschriften entspricht.
- [AWS Security Hub](#) – Dieser AWS-Service liefert einen umfassenden Überblick über den Sicherheitsstatus in AWS. So können Sie die Compliance mit den Sicherheitsstandards in der Branche und den bewährten Methoden abgleichen.

Eine Liste der AWS-Services, die in den Geltungsbereich bestimmter Compliance-Programme fallen, finden Sie auf der Seite [AWS-Services in Scope nach Compliance-Programm](#). Allgemeine Informationen finden Sie unter [AWS-Compliance-Programme](#).

## Ausfallsicherheit in Amazon Comprehend

Die AWS globale -Infrastruktur ist um AWS-Region en und Availability Zones herum aufgebaut. AWS-Region en bieten mehrere physisch getrennte und isolierte Availability Zones, die mit einem Netzwerk mit niedriger Latenz, hohem Durchsatz und hoher Redundanz verbunden sind. Mithilfe von Availability Zones können Sie Anwendungen und Datenbanken erstellen und ausführen, die automatisch Failover zwischen Availability Zones ausführen, ohne dass es zu Unterbrechungen kommt. Availability Zones sind besser hoch verfügbar, fehlertoleranter und skalierbarer als herkömmliche Infrastrukturen mit einem oder mehreren Rechenzentren.

Weitere Informationen zu AWS-Region en und Availability Zones finden Sie unter [AWS Globale Infrastruktur](#).

## Infrastruktursicherheit in Amazon Comprehend

Als verwalteter Service ist Amazon Comprehend durch die AWS globale Netzwerksicherheit von geschützt. Informationen zu AWS-Sicherheitsservices und wie AWS die Infrastruktur schützt, finden Sie unter [AWS-Cloud-Sicherheit](#). Informationen zum Entwerfen Ihrer AWS Umgebung mit den bewährten Methoden für die Infrastruktursicherheit finden Sie unter [Infrastrukturschutz](#) in Security Pillar AWS Well-Architected Framework.

Sie verwenden durch AWS veröffentlichte API-Aufrufe, um über das Netzwerk auf Amazon Comprehend zuzugreifen. Kunden müssen Folgendes unterstützen:

- Transport Layer Security (TLS). Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Verschlüsselungs-Suiten mit Perfect Forward Secrecy (PFS) wie DHE (Ephemeral Diffie-Hellman) oder ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Die meisten modernen Systeme wie Java 7 und höher unterstützen diese Modi.

Außerdem müssen Anforderungen mit einer Zugriffsschlüssel-ID und einem geheimen Zugriffsschlüssel signiert sein, der einem IAM-Prinzipal zugeordnet ist. Alternativ können Sie mit [AWS Security Token Service](#) (AWS STS) temporäre Sicherheitsanmeldeinformationen erstellen, um die Anforderungen zu signieren.

# Richtlinien und Kontingente

Sofern nicht anders angegeben, gelten die Amazon Comprehend Comprehend-Kontingente pro Region. Sie können eine Erhöhung der anpassbaren Kontingente beantragen, falls dies für Ihre Anwendungen erforderlich ist. Informationen zu Kontingenten und zur Beantragung einer Kontingenterhöhung finden Sie unter [AWS Service Quotas](#).

## Themen

- [Unterstützte Regionen](#)
- [Kontingente für integrierte Modelle](#)
- [Kontingente für benutzerdefinierte Modelle](#)
- [Kontingente für Schwungräder](#)

## Unterstützte Regionen

Amazon Comprehend ist in den folgenden AWS Regionen verfügbar:

- US East (Ohio)
- USA Ost (Nord-Virginia)
- USA West (Oregon)
- Asia Pacific (Mumbai)
- Asia Pacific (Seoul)
- Asien-Pazifik (Singapur)
- Asien-Pazifik (Sydney)
- Asien-Pazifik (Tokio)
- Canada (Central)
- Europe (Frankfurt)
- Europa (Irland)
- Europe (London)
- AWS GovCloud (US-West)

Standardmäßig bietet Amazon Comprehend alle API-Operationen in jeder der unterstützten Regionen. Ausnahmen finden Sie unter [Dokumentenverarbeitung](#)

Informationen zu API-Endpunkten finden Sie unter [Amazon Comprehend Regions and Endpoints](#) in der Amazon Web Services General Reference.

Um die aktuellen Kontingente in einer Region zu überprüfen oder um Kontingenterhöhungen für anpassbare Kontingente zu beantragen, öffnen Sie die [Service-Kontingents-Konsole](#).

## Kontingente für integrierte Modelle

Amazon Comprehend bietet integrierte Modelle für die Analyse von UTF-8-Textdokumenten. Amazon Comprehend bietet synchrone und asynchrone Operationen, die die integrierten Modelle verwenden.

Themen

- [\(synchrone\) Analyse in Echtzeit](#)
- [Asynchrone Analyse](#)

### (synchrone) Analyse in Echtzeit

In diesem Abschnitt werden die Kontingente im Zusammenhang mit Echtzeitanalysen unter Verwendung der integrierten Modelle beschrieben.

Themen

- [Operationen für einzelne Dokumente](#)
- [Operationen mit mehreren Dokumenten](#)
- [Anforderungsdrosselung für \(synchrone\) Anfragen in Echtzeit](#)

### Operationen für einzelne Dokumente

Die Amazon Comprehend API bietet Operationen, die ein einzelnes Dokument als Eingabe verwenden. Die folgenden Kontingente gelten für diese Operationen.

Allgemeine Kontingente für Operationen mit nur einem Dokument

Die folgenden Kontingente gelten für Echtzeitanalysen zur Erkennung von Entitäten, Schlüsselwörtern oder dominanter Sprache. Für die Erkennung von Entitäten gelten diese Kontingente für die Erkennung mit den integrierten Modellen. Informationen zur benutzerdefinierten Entitätserkennung finden Sie in den Kontingenten unter [Benutzerdefinierte Entitätserkennung](#).

Beschreibung	Kontingent/Richtlinie
Maximale Dokumentgröße	100 KB

### Vorgangsspezifische Kontingente für Operationen mit einzelnen Dokumenten

Die folgenden Kontingente gelten für Echtzeitanalysen zur Erkennung von Stimmungen, gezielter Stimmungen und Syntax.

Beschreibung	Quot/Richtlinie
Maximale Dokumentgröße	5 KB

### Operationen mit mehreren Dokumenten

Die Amazon Comprehend API bietet Batch-Operationen, die mehrere Dokumente mit einer einzigen API-Anfrage verarbeiten. Die folgenden Kontingente gelten für die Batch-Operationen.

Beschreibung	Kontingent/Richtlinie
Maximale Dokumentgröße	5 KB
Maximale Anzahl Dokumente pro Anfrage	25

Weitere Hinweise zur Verwendung der Batch-Dokumentoperationen finden Sie unter [Synchrone Verarbeitung mehrerer Dokumente](#).

### Anforderungsdrosselung für (synchrone) Anfragen in Echtzeit

Amazon Comprehend wendet dynamische Drosselung auf synchrone Anfragen an. Wenn Bandbreite für die Systemverarbeitung verfügbar ist, erhöht Amazon Comprehend schrittweise die Anzahl Ihrer Anfragen, die es verarbeitet. Um die Nutzung der synchronen API-Operationen durch Ihre Anwendung zu kontrollieren, empfehlen wir Ihnen, Abrechnungswarnungen zu aktivieren oder eine Gebührenbegrenzung in Ihrer Anwendung zu implementieren.

## Asynchrone Analyse

In diesem Abschnitt werden Kontingente im Zusammenhang mit asynchroner Analyse unter Verwendung der integrierten Modelle beschrieben.

Asynchrone API-Operationen unterstützen jeweils maximal 10 aktive Jobs. Die Kontingente für jeden API-Vorgang finden Sie in der Tabelle Service [Quotas in Amazon Comprehend Endpoints und Kontingente](#) in der Amazon Web Services General Reference.

Für anpassbare Kontingente können Sie über die [Service Quotas-Konsole eine Erhöhung des Kontingents](#) beantragen.

### Themen

- [Allgemeine Kontingente für asynchrone Operationen](#)
- [Vorgangsspezifische Kontingente für asynchrone Jobs](#)
- [Drosselung von Anfragen für asynchrone Anfragen](#)

### Allgemeine Kontingente für asynchrone Operationen

Sie können asynchrone Analysejobs mithilfe der Konsole oder einer der Start\* API-Operationen ausführen. Hinweise dazu, wann asynchrone Operationen verwendet werden sollten, finden Sie unter [Asynchrone Batchverarbeitung](#). Die folgenden Kontingente gelten für die meisten Start\* API-Operationen für integrierte Modelle. Die Ausnahmen finden Sie unter [Vorgangsspezifische Kontingente für asynchrone Jobs](#).

Beschreibung	Kontingent/Richtlinie
Maximale Größe jedes Dokuments in Jobs, bei denen Entitäten, Schlüsselausdrücke, personenbezogene Daten und Sprachen erkannt werden	1 MB
Maximale Gesamtgröße aller Dateien in einer Anfrage	5 GB
Minimale Gesamtgröße aller Dateien in einer Anfrage	500 Byte
Maximale Anzahl von Dateien, ein Dokument pro Datei	1 000 000
Maximale Gesamtzahl der Zeilen, ein Dokument pro Zeile	1 000 000



## Vorgangsspezifische Kontingente für asynchrone Jobs

In diesem Abschnitt werden Kontingente für bestimmte asynchrone Operationen beschrieben. Wenn in den folgenden Tabellen kein Kontingent angegeben ist, gilt der allgemeine Kontingentwert.

### Themen

- [Stimmung](#)
- [Zielgerichtete Stimmung](#)
- [Ereignisse](#)
- [Themenmodellierung](#)

### Stimmung

Asynchrone Sentiment-Jobs, die Sie mit dem [StartSentimentDetectionJob](#) Vorgang erstellen, haben die folgenden Kontingente.

Beschreibung	Kontingent/Richtlinie
Maximale Größe jedes Eingabedokuments	5 KB

### Zielgerichtete Stimmung

Asynchrone, gezielte Stimmungsjobs, die Sie mit dem [StartTargetedSentimentDetectionJob](#) Vorgang erstellen, haben die folgenden Kontingente.

Beschreibung	Kontingent/Richtlinie
Unterstützte Dokumentenformate	UTF-8
Maximale Größe jedes Dokuments in einem Job	10 KB
Maximale Größe aller Dokumente in einem Job	300 MB
Maximale Anzahl von Dateien, ein Dokument pro Datei	30 000
Maximale Gesamtzahl von Zeilen, ein Dokument pro Zeile (für alle Dateien in einer Anfrage)	30 000

## Ereignisse

Für Aufträge zur Erkennung asynchroner Ereignisse, die Sie mit dem [StartEventsDetectionJob](#) Vorgang erstellen, gelten die folgenden Kontingente.

Beschreibung	Kontingente
Zeichencodierung	UTF-8
Gesamtgröße aller Dateien in einem Auftrag	50 MB
Maximale Größe jedes Dokuments in einem Job	10 KB
Maximale Anzahl von Dateien, ein Dokument pro Datei	5,000
Maximale Gesamtzahl der Zeilen, ein Dokument pro Zeile (für alle angeforderten Dateien)	5,000

## Themenmodellierung

Asynchrone Themenmodellierungsaufträge, die Sie mit dem [StartTopicsDetectionJob](#) Vorgang erstellen, haben die folgenden Kontingente.

Beschreibung	Kontingent/Richtlinie
Zeichencodierung	UTF-8
Maximale Anzahl von Themen, die zurückgegeben werden sollen	100
Maximale Dateigröße für eine Datei, ein Dokument pro Datei	100 MB

Weitere Informationen finden Sie unter [Themenmodellierung](#).

## Drosselung von Anfragen für asynchrone Anfragen

Jeder asynchrone API-Vorgang unterstützt eine maximale Anzahl von Anfragen pro Sekunde (pro Region, pro Konto) sowie maximal 10 aktive Jobs. Die Kontingente für jeden API-Vorgang finden Sie

in der Tabelle Service [Quotas in Amazon Comprehend Endpoints und Kontingente](#) in der Amazon Web Services General Reference.

Für anpassbare Kontingente können Sie über die [Service Quotas-Konsole eine Erhöhung des Kontingents](#) beantragen.

## Kontingente für benutzerdefinierte Modelle

Sie können Amazon Comprehend verwenden, um Ihre eigenen benutzerdefinierten Modelle für die benutzerdefinierte Klassifizierung und benutzerdefinierte Entitätserkennung zu erstellen. Dieser Abschnitt enthält die Richtlinien und Kontingente für Schulungen und die Verwendung benutzerdefinierter Modelle. Weitere Informationen zu benutzerdefinierten Modellen finden Sie unter [Amazon Comprehend Custom](#).

Themen

- [Allgemeine Kontingente](#)
- [Kontingente für Endgeräte](#)
- [Klassifizierung von Dokumenten](#)
- [Benutzerdefinierte Entitätserkennung](#)

## Allgemeine Kontingente

Amazon Comprehend legt allgemeine Größenkontingente für jeden Typ von Eingabedokument fest, die Sie mit benutzerdefinierten Modellen analysieren können. Informationen zu Quoten für Analysen in Echtzeit finden Sie unter [Maximale Dokumentgrößen für Echtzeitanalysen](#). Informationen zu asynchronen Analysequoten finden Sie unter [Eingaben für asynchrone benutzerdefinierte Analysen](#).

Jeder asynchrone API-Vorgang unterstützt eine maximale Anzahl von Anfragen pro Sekunde (pro Region, pro Konto) sowie maximal 10 aktive Jobs. Die Kontingente für jeden API-Vorgang finden Sie in der Tabelle Service [Quotas in Amazon Comprehend Endpoints und Kontingente](#) in der Amazon Web Services General Reference.

Für anpassbare Kontingente können Sie über die [Service Quotas-Konsole eine Erhöhung des Kontingents](#) beantragen.

## Kontingente für Endgeräte

Sie erstellen einen Endpunkt, um Echtzeitanalysen mit einem benutzerdefinierten Modell durchzuführen. Informationen zu Endpunkten finden Sie unter [Verwalten von Amazon Comprehend-Endpunkten](#).

Die folgenden Kontingente gelten für die Endgeräte. Informationen dazu, wie Sie eine Kontingenterhöhung beantragen können, finden Sie unter [AWS Service Quotas](#).

Beschreibung	Kontingent/Richtlinie
Maximale Anzahl aktiver Endpunkte pro Region für jedes Konto	20
Maximale Anzahl von Inferenzeinheiten pro Region für jedes Konto	200
Maximale Anzahl von Inferenzeinheiten pro Endpunkt pro Region	50
Maximaler Durchsatz pro Inferenzeinheit (Zeichen)	100/Sekunde
Maximaler Durchsatz pro Inferenzeinheit (Dokumente)	2/Sekunde

## Klassifizierung von Dokumenten

In diesem Abschnitt werden die Richtlinien und Kontingente für die folgenden Verfahren zur Klassifizierung von Dokumenten beschrieben:

- Classifier-Schulungsaufgaben, die Sie mit der [CreateDocumentClassifier](#) Operation beginnen.
- Asynchrone Aufträge zur Dokumentenklassifizierung, die Sie mit der [StartDocumentClassificationJob](#) Operation starten.
- Synchrone Dokumentklassifizierungsanforderungen, die den [ClassifyDocument](#) Vorgang verwenden.

## Allgemeine Kontingente für die Klassifizierung von Dokumenten

In der folgenden Tabelle werden allgemeine Kontingente für das Training benutzerdefinierter Klassifikatoren beschrieben.

Beschreibung	Kontingent/Richtlinie
Maximale Länge des Klassennamens	5.000 Zeichen
Anzahl der Klassen (Mehrklassenmodus)	2—1.000
Anzahl der Klassen (Multi-Label-Modus)	2—100
Format der Anmerkungen	
Mindestanzahl von Anmerkungen pro Klasse (Mehrklassenmodus)	10
Mindestanzahl von Anmerkungen pro Klasse (Multi-Label-Modus)	10
Mindestanzahl von Anmerkungen (Multi-Label-Modus)	50
CSV-Dateiformat	
Mindestanzahl von Schulungsdokumenten pro Kurs (Mehrklassenmodus)	50
Mindestanzahl von Schulungsdokumenten pro Kurs (Modus mit mehreren Bezeichnungen)	10
Mindestanzahl von Schulungsdokumenten (Modus mit mehreren Labels)	50

## Klassifizierung für Klartextdokumente

Sie erstellen und trainieren ein Klartext-Modell mithilfe von Eingabedokumenten im Klartext-Format. Amazon Comprehend bietet asynchrone Echtzeitoperationen zur Klassifizierung von Klartext-Dokumenten mithilfe eines Klartext-Modells.

### Training

In der folgenden Tabelle werden die Kontingente für das Training eines benutzerdefinierten Klassifikators mit Klartextdokumenten beschrieben.

Beschreibung	Kontingent/Richtlinie
Gesamtgröße aller Dateien im Trainingsjob	5 GB
Maximale Anzahl erweiterter Manifestdateien für das Training eines benutzerdefinierten Klassifikators	5
Maximale Anzahl von Attributnamen für jede erweiterte Manifestdatei	5
Maximale Länge des Attributnamens	63 Zeichen

### (synchrone) Analyse in Echtzeit

In der folgenden Tabelle werden die Kontingente für die Echtzeitklassifizierung von Klartextdokumenten beschrieben.

Beschreibung	Kontingent/Richtlinie
Maximale Anzahl von Dokumenten pro synchroner Anfrage	1
Maximale Größe von Textdokumenten (UTF-8-kodiert)	10 KB

### Asynchrone Analyse

In der folgenden Tabelle werden die Kontingente für die asynchrone Klassifizierung von Klartextdokumenten beschrieben.

Beschreibung	Kontingent/Richtlinie
Gesamtgröße aller Dateien im asynchronen Job	5 GB
Maximale Dateigröße für eine Datei, ein Dokument pro Datei	10 MB
Maximale Anzahl von Dateien, ein Dokument pro Datei	1 000 000
Maximale Gesamtzahl der Zeilen, ein Dokument pro Zeile (für alle angeforderten Dateien)	1 000 000

## Klassifizierung für halbstrukturierte Dokumente

In diesem Abschnitt werden die Richtlinien und Quoten für die Klassifizierung halbstrukturierter Dokumente beschrieben. Verwenden Sie zur Klassifizierung halbstrukturierter Dokumente ein systemeigenes Dokumentenmodell, das Sie mit systemeigenen Eingabedokumenten trainiert haben.

Trainieren eines systemeigenen Dokumentenmodells mit halbstrukturierten Dokumenten

In der folgenden Tabelle werden die Kontingente für das Training eines benutzerdefinierten Klassifikators mit halbstrukturierten Dokumenten wie PDF-Dokumenten, Word-Dokumenten und Bilddateien beschrieben.

Beschreibung	Kontingent/Richtlinie
Maximale Seitenanzahl für alle Dokumente	10.000
Maximale Dateigröße für Anmerkungen (alle CSV-Dateigrößen zusammen)	5 MB
Größe des Dokumentkorpus (Schulungs- und Testdokumente)	10 GB
Dateigrößen für Trainings- und Testdateien	
Größe der Bilddatei (JPG, PNG, TIFF).	1 Byte — 10 MB.  TIFF-Dateien: maximal eine Seite.
Seitengröße für PDF-Dokumente	1 Byte — 10 MB
Seitengröße für Word-Dokumente	1 Byte — 10 MB
JSON-Ausgabegröße der Amazon Textract Textract-API	1 Byte — 1 MB

(synchrone) Analyse in Echtzeit

In diesem Abschnitt werden Quoten für die Echtzeitklassifizierung von halbstrukturierten Dokumenten beschrieben.

Die folgende Tabelle zeigt die maximalen Dateigrößen für Eingabedokumente. Für alle Eingabedokumententypen beträgt das Maximum der Eingabedatei eine Seite mit nicht mehr als 10.000 Zeichen.

Dateityp	Maximale Größe (API)	Maximale Größe (Konsole)
UTF-8-Textdokumente	10 KB	10 KB
PDF-Dokumente	10 MB	5 MB
Word-Dokumente	10 MB	5 MB
Abbildungsdateien	10 MB	5 MB
Ausgabegröße der Amazon Textract Textract-API	1 MB	–

### Asynchrone Analyse

In der folgenden Tabelle werden die Kontingente für die asynchrone Klassifizierung halbstrukturierter Dokumente beschrieben.

Beschreibung	Kontingent/Richtlinie
Maximale Seitenanzahl aller Eingabedokumente für einen Job	25,000
Größe des Dokumentkorpus	25 GB
Größe der Bilddatei (JPG, PNG oder TIFF)	1 Byte — 10 MB.  TIFF-Dateien: maximal eine Seite.
Seitengröße für PDF-Dokumente	1 Byte — 10 MB
Seitengröße für Word-Dokumente	1 Byte — 10 MB
JSON-Ausgabegröße der Textract-API	1 Byte — 1 MB.



## Benutzerdefinierte Entitätserkennung

In diesem Abschnitt werden die Richtlinien und Kontingente für die folgenden Operationen zur benutzerdefinierten Entitätserkennung beschrieben:

- Die Trainingsaufträge für Entity Recognizer wurden mit dem [CreateEntityRecognizer](#) Vorgang gestartet.
- Asynchrone Entitätserkennungsaufträge wurden mit dem [StartEntitiesDetectionJob](#) Vorgang gestartet.
- Synchrone Anfragen zur Entitätserkennung, die den [DetectEntities](#) Vorgang verwenden.

### Benutzerdefinierte Entitätserkennung für Klartextdokumente

Amazon Comprehend bietet asynchrone und synchrone Operationen zur Analyse von Klartextdokumenten mit einer benutzerdefinierten Entitätserkennung.

#### Training

In diesem Abschnitt werden die Kontingente für das Training eines benutzerdefinierten Entity Recognizers zur Analyse von Klartext-Dokumenten beschrieben. Um das Modell zu trainieren, können Sie eine Entitätsliste oder einen Satz kommentierter Textdokumente bereitstellen.

In der folgenden Tabelle werden die Kontingente für das Trainieren des Modells mit einer Entitätsliste beschrieben.

Beschreibung	Kontingent/Richtlinie
Anzahl der Einheiten pro Modell	1—25
Größe des Dokuments (UTF-8)	1—5.000 Byte
Anzahl der Elemente in der Entitätsliste	1—1 Million
Länge des einzelnen Eintrags (Post-Strip) in der Eintragsliste	1—5.000
Korpusgröße der Entitätsliste (alle Dokumente im Klartext zusammengefasst)	5 KB —200 MB

In der folgenden Tabelle werden die Kontingente für das Training des Modells mit kommentierten Textdokumenten beschrieben.

Beschreibung	Kontingent/Richtlinie
Anzahl der Entitäten pro Modell/benutzerdefinierter Entitätenerkennung	1—25
Größe des Dokuments (UTF-8)	1—5.000 Byte
Anzahl der Dokumente (siehe <a href="#">Klartext-Anmerkungen</a> )	3—200.000
Größe des Dokumentkorpus (alle Dokumente im Klartext-Format zusammen)	5 KB - 200 MB
Mindestanzahl von Anmerkungen pro Entität	25

#### (synchroner) Analyse in Echtzeit

In der folgenden Tabelle werden die Kontingente für die Echtzeitanalyse von Klartextdokumenten beschrieben.

Beschreibung	Kontingent/Richtlinie
Maximale Anzahl von Dokumenten pro synchroner Anfrage	1
Maximale Größe von Textdokumenten (UTF-8-kodiert)	5 KB

#### Asynchrone Analyse

In der folgenden Tabelle werden die Kontingente für die asynchrone Entitätserkennung von Klartextdokumenten beschrieben.

Beschreibung	Kontingent/Richtlinie
Größe des Dokuments (UTF-8)	1 Byte — 1 MB
Maximale Anzahl von Dateien, ein Dokument pro Datei	1 000 000

Beschreibung	Kontingent/Richtlinie
Maximale Gesamtzahl der Zeilen, ein Dokument pro Zeile (für alle angeforderten Dateien)	1 000 000
Korpusgröße des Dokuments (alle Dokumente im Klartext zusammengefasst)	1 Byte — 5 GB

## Benutzerdefinierte Entitätserkennung für halbstrukturierte Dokumente

Amazon Comprehend bietet asynchrone und synchrone Operationen zur Analyse halbstrukturierter Dokumente mit einer benutzerdefinierten Entitätserkennung. Sie müssen das Modell mithilfe von kommentierten PDF-Dokumenten trainieren.

### Training

In der folgenden Tabelle werden die Kontingente für das Training eines benutzerdefinierten Entity Recognizers (CreateEntityRecognizer) zur Analyse halbstrukturierter Dokumente beschrieben.

Beschreibung	Kontingent/Richtlinie
Anzahl der Entitäten pro Modell/benutzerdefinierter Entitätenerkennung	1—25
Maximale Größe der Annotationsdatei (UTF-8 JSON)	5 MB
Anzahl der Dokumente	250—10.000
Größe des Dokumentkorpus (alle Dokumente im Klartext-Format zusammen)	5 KB — 1 GB
Mindestanzahl von Anmerkungen pro Entität	100
Maximale Anzahl erweiterter Manifestdateien zum Trainieren eines benutzerdefinierten Entitätserkenners	5
Maximale Anzahl von Attributnamen für jede erweiterte Manifestdatei	5

Beschreibung	Kontingent/Richtlinie
Maximale Länge des Attributnamens	63 Zeichen

### (synchrone) Analyse in Echtzeit

In diesem Abschnitt werden die Kontingente für die Echtzeitanalyse halbstrukturierter Dokumente beschrieben.

Die folgende Tabelle zeigt die maximalen Dateigrößen für Eingabedokumente. Für alle Eingabedokumenttypen beträgt das Maximum der Eingabedatei eine Seite mit nicht mehr als 10.000 Zeichen.

Dateityp	Maximale Größe (API)	Maximale Größe (Konsole)
UTF-8-Textdokumente	10 KB	10 KB
PDF-Dokumente	10 MB	5 MB
Word-Dokumente	10 MB	5 MB
Abbildungsdateien	10 MB	5 MB
Textausgabedateien	1 MB	–

### Asynchrone Analyse

In diesem Abschnitt werden Kontingente für die asynchrone Analyse halbstrukturierter Dokumente beschrieben.

Beschreibung	Kontingent/Richtlinie
Bildgröße (JPG oder PNG)	1 Byte — 10 MB
Bildgröße (TIFF)	1 Byte — 10 MB. Maximal eine Seite.

Beschreibung	Kontingent/Richtlinie
Größe des Dokuments (PDF)	1 Byte — 50 MB
Größe des Dokuments (Docx)	1 Byte — 5 MB
Größe des Dokuments (UTF-8)	1 Byte — 1 MB
Maximale Anzahl von Dateien, ein Dokument pro Datei (ein Dokument pro Zeile ist für Bilddateien oder PDF-/Word-Dokumente nicht zulässig)	500
Maximale Anzahl von Seiten für eine PDF- oder Docx-Datei	100
Korpusgröße des Dokuments nach der Textextraktion (Klartext, alle Dateien zusammen)	1 Byte — 5 GB

Weitere Informationen zu Grenzwerten für Bilder finden Sie unter [Harte Grenzwerte in Amazon Textract](#)

## Kontingente für Schwungräder

Verwenden Sie Schwungräder, um das Training und die Nachverfolgung benutzerdefinierter Modellversionen für die benutzerdefinierte Klassifizierung und Erkennung benutzerdefinierter Entitäten zu verwalten. Weitere Informationen zu Schwungrädern finden Sie unter [Flywheels](#)

### Allgemeine Kontingente für Schwungräder

Die folgenden Kontingente gelten für Schwungräder und Schwungrad-Iterationen.

Beschreibung	Kontingent/Richtlinie
Maximale Anzahl von Schwungrädern	50
Maximale Anzahl von Schwungrädern im Status CREATING	10
Maximale Anzahl von Trainingsdatensätzen pro Schwungrad	50
Maximale Anzahl von Testdatensätzen pro Schwungrad	50

Beschreibung	Kontingent/Richtlinie
Maximale Anzahl von Datensätzen mit dem Status INGESTING	10
Maximale Anzahl laufender Schwungraditerationen pro Konto	10

## Datensatzquoten für benutzerdefinierte Klassifizierungsmodelle

Wenn Sie einen Datensatz für ein Schwungrad aufnehmen, das einem benutzerdefinierten Klassifikationsmodell zugeordnet ist, gelten die folgenden Kontingente.

Beschreibung	Kontingent/Richtlinie
Mindestanzahl von Schulungsdokumenten pro Kurs (Modus mit mehreren Labels)	50
Maximale Anzahl von Schulungsdokumenten	1 000 000
Minimale Datensatzgröße	500 Byte
Maximale Datensatzgröße	5 GB
Maximale Dateigröße für eine Datei, ein Dokument pro Datei	10 MB

## Datensatzquoten für benutzerdefinierte Modelle zur Erkennung von Entitäten

Wenn Sie einen Datensatz für ein Schwungrad aufnehmen, das einem benutzerdefinierten Entitätserkennungsmodell zugeordnet ist, gelten die folgenden Kontingente.

Beschreibung	Kontingent/Richtlinie
Maximale Dokumentgröße	5 KB
Mindestanzahl von Schulungsunterlagen	3
Maximale Anzahl von Schulungsdokumenten	200 000

Beschreibung	Kontingent/Richtlinie
Mindestanzahl von Anmerkungen pro Entität	25
Maximale Datensatzgröße	200 MB

# Tutorials und andere Ressourcen

Tutorials und andere Ressourcen für Amazon Comprehend .

Themen

- [Tutorial: Analysieren von Erkenntnissen aus Kundenrezensionen mit Amazon Comprehend](#)
- [Verwenden von Amazon S3-Objekt-Lambda-Zugriffspunkten für persönlich identifizierbare Informationen \(PII\)](#)
- [Lösung: Analysieren von Text mit Amazon Comprehend und OpenSearch](#)

## Tutorial: Analysieren von Erkenntnissen aus Kundenrezensionen mit Amazon Comprehend

In diesem Tutorial wird erläutert, wie Sie Amazon Comprehend mit [Amazon Simple Storage Service](#) , [AWS Glue](#), und [Amazon QuickSight](#) verwenden [Amazon Athena](#), um wertvolle Einblicke in Ihre Dokumente zu erhalten. Amazon Comprehend kann Stimmungen (die Stimmung eines Dokuments) und Entitäten (Namen von Personen, Organisationen, Ereignissen, Datumsangaben, Produkten, Orten, Mengen und Titeln) aus unstrukturiertem Text extrahieren.

Sie können beispielsweise verwertbare Erkenntnisse aus Kundenrezensionen erhalten. In diesem Tutorial analysieren Sie einen Beispieldatensatz mit Kundenrezensionen zu einem Neuzugang. Sie verwenden die Amazon Comprehend-Emotionsanalyse, um festzustellen, ob sich Kunden positiv oder negativ gegenüber dem Neuzugang fühlen. Sie verwenden auch die Amazon Comprehend-Entitätsanalyse, um Erwähnungen wichtiger Entitäten zu entdecken, z. B. verwandte Neuerungen oder Autoren. Nach diesem Tutorial stellen Sie möglicherweise fest, dass über 50 % der Bewertungen positiv sind. Möglicherweise stellen Sie auch fest, dass Kunden Autoren vergleichen und Interesse an anderen klassischen Neuerungen ausdrücken.

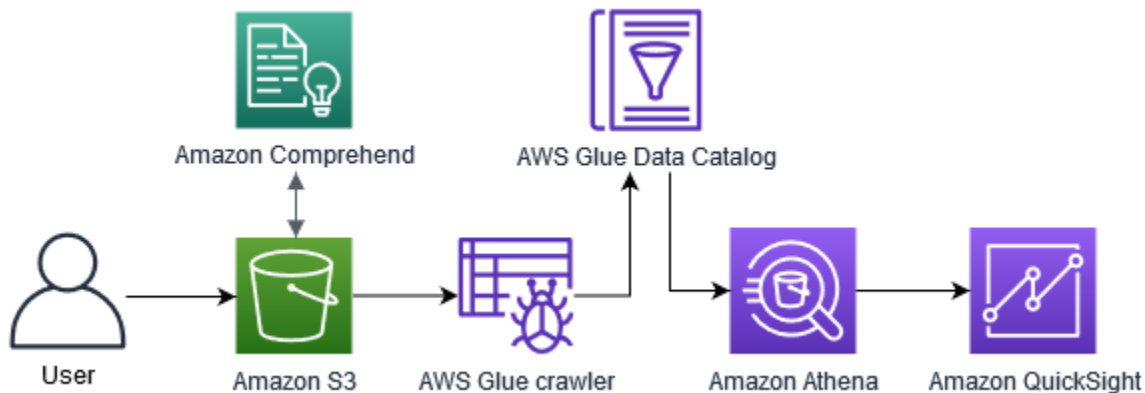
In diesem Tutorial führen Sie Folgendes aus:

- Speichern Sie einen Beispieldatensatz mit Überprüfungen in [Amazon Simple Storage Service](#) (Amazon S3). Amazon Simple Storage Service ist ein Objektspeicherservice.
- Verwenden Sie [Amazon Comprehend](#), um die Stimmung und Entitäten in den Bewertungsdokumenten zu analysieren.



- Verwenden Sie einen [AWS Glue](#) -Crawler, um die Ergebnisse der Analyse in einer Datenbank zu speichern. AWS Glue ist ein Extract, Transform, Load (ETL)-Service, mit dem Sie Ihre Daten für Analysen katalogisieren und bereinigen können.
- Führen Sie [Amazon Athena](#) Abfragen aus, um Ihre Daten zu bereinigen. Amazon Athena ist ein interaktiver Serverless-Abfrageservice.
- Erstellen Sie Visualisierungen mit Ihren Daten in [Amazon QuickSight](#). Amazon QuickSight ist ein Serverless-Business-Intelligence-Tool zum Extrahieren von Erkenntnissen aus Ihren Daten.

Das folgende Diagramm zeigt den Workflow.



Geschätzte Zeit für die Durchführung dieses Tutorials: 1 Stunde

Geschätzte Kosten: Für einige der Aktionen in diesem Tutorial fallen Gebühren für Ihr anAWS-Konto. Informationen zu den Gebühren für jeden dieser Services finden Sie auf den folgenden Preisseiten.

- [Amazon S3 – Preise](#)
- [Amazon Comprehend – Preise](#)
- [AWS Glue-Preise](#)
- [Amazon Athena-Preise](#)
- [Amazon QuickSight-Preise](#)

## Themen

- [Voraussetzungen](#)
- [Schritt 1: Dokumente zu Amazon S3 hinzufügen](#)
- [Schritt 2: \(nur CLI\) Erstellen einer IAM-Rolle für Amazon Comprehend](#)
- [Schritt 3: Ausführen von Analyseaufträgen für Dokumente in Amazon S3](#)

- [Schritt 4: Vorbereiten der Amazon Comprehend für die Datenvisualisierung](#)
- [Schritt 5: Visualisieren der Amazon Comprehend-Ausgabe in Amazon QuickSight](#)

## Voraussetzungen

Zum Durcharbeiten dieses Tutorials ist Folgendes erforderlich:

- Ein(e) AWS-Konto. Informationen zum Einrichten eines finden AWS-KontoSie unter [Einrichtung](#).
- Eine IAM-Entität (Benutzer, Gruppe oder Rolle). Informationen zum Einrichten von Benutzern und Gruppen für Ihr -Konto finden Sie im Tutorial [Erste Schritte](#) im IAM-Benutzerhandbuch.
- Die folgende Berechtigungsrichtlinie, die Ihrem Benutzer, Ihrer Gruppe oder Ihrer Rolle zugeordnet ist. Die Richtlinie gewährt einige der Berechtigungen, die für das Abschließen dieses Tutorials erforderlich sind. Die nächste Voraussetzung beschreibt die zusätzlichen Berechtigungen, die Sie benötigen.

```
{
  "Version": "2012-10-17",
  "Statement":
  [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action":
      [
        "comprehend:*",
        "ds:AuthorizeApplication",
        "ds:CheckAlias",
        "ds:CreateAlias",
        "ds:CreateIdentityPoolDirectory",
        "ds>DeleteDirectory",
        "ds:DescribeDirectories",
        "ds:DescribeTrusts",
        "ds:UnauthorizeApplication",
        "iam:AttachRolePolicy",
        "iam:CreatePolicy",
        "iam:CreatePolicyVersion",
        "iam:CreateRole",
        "iam>DeletePolicyVersion",
        "iam>DeleteRole",
        "iam:DetachRolePolicy",
```

```

    "iam:GetPolicy",
    "iam:GetPolicyVersion",
    "iam:GetRole",
    "iam:ListAccountAliases",
    "iam:ListAttachedRolePolicies",
    "iam:ListEntitiesForPolicy",
    "iam:ListPolicies",
    "iam:ListPolicyVersions",
    "iam:ListRoles",
    "quicksight:*",
    "s3:*",
    "tag:GetResources"
  ],
  "Resource": "*"
},
{
  "Action":
  [
    "iam:PassRole"
  ],
  "Effect": "Allow",
  "Resource":
  [
    "arn:aws:iam::*:role/*Comprehend*"
  ]
}
]
}

```

Verwenden Sie die vorherige Richtlinie, um eine IAM-Richtlinie zu erstellen und sie an Ihre Gruppe oder Ihren Benutzer anzuhängen. Informationen zum Erstellen einer IAM-Richtlinie finden Sie unter [Erstellen von IAM-Richtlinien](#) im IAM-Benutzerhandbuch. Informationen zum Anfügen einer IAM-Richtlinie finden Sie unter [Hinzufügen und Entfernen von IAM-Identitätsberechtigungen](#) im IAM-Benutzerhandbuch.

- Verwaltete Richtlinien, die an Ihre IAM-Gruppe oder Ihren IAM-Benutzer angefügt sind. Zusätzlich zur vorherigen Richtlinie müssen Sie Ihrer Gruppe oder Ihrem Benutzer auch die folgenden AWS verwalteten Richtlinien anfügen:
  - AWSGlueConsoleFullAccess
  - AWSQuicksightAthenaAccess

Diese verwalteten Richtlinien erteilen Ihnen die Berechtigung zur Verwendung AWS Glue von Amazon Athena, und Amazon QuickSight. Informationen zum Anfügen einer IAM-Richtlinie finden Sie unter [Hinzufügen und Entfernen von IAM-Identitätsberechtigungen](#) im IAM-Benutzerhandbuch.

## Schritt 1: Dokumente zu Amazon S3 hinzufügen

Bevor Sie mit den Amazon Comprehend Comprehend-Analysejobs beginnen, müssen Sie einen Beispieldatensatz mit Kundenrezensionen in Amazon Simple Storage Service (Amazon S3) speichern. Amazon S3 hostet Ihre Daten in Containern, sogenannten Buckets. Amazon Comprehend kann Dokumente analysieren, die in einem Bucket gespeichert sind, und sendet die Ergebnisse der Analyse an einen Bucket. In diesem Schritt erstellen Sie einen S3-Bucket, erstellen Eingabe- und Ausgabeordner im Bucket und laden einen Beispieldatensatz in den Bucket hoch.

### Themen

- [Voraussetzungen](#)
- [Laden Sie Beispieldaten herunter](#)
- [Erstellen eines Amazon-S3-Buckets](#)
- [\(nur Konsole\) Ordner erstellen](#)
- [Laden Sie die Eingabedaten hoch](#)

### Voraussetzungen

Bevor Sie beginnen, überprüfen [Tutorial: Analysieren von Erkenntnissen aus Kundenrezensionen mit Amazon Comprehend](#) und erfüllen Sie die Voraussetzungen.

### Laden Sie Beispieldaten herunter

Der folgende Beispieldatensatz enthält Amazon-Rezensionen aus dem größeren Datensatz „Amazon reviews — Full“, der zusammen mit dem Artikel „Character-level Convolutional Networks for Text Classification“ (Xiang Zhang et al., 2015) veröffentlicht wurde. Laden Sie den Datensatz auf Ihren Computer herunter.

Um die Beispieldaten zu erhalten

1. Laden Sie die ZIP-Datei ([tutorial-reviews-data.zip](#)) auf Ihren Computer herunter.

2. Extrahieren Sie die Zip-Datei auf Ihrem Computer. Es gibt zwei Dateien. Die Datei `THIRD_PARTY_LICENSES.txt` ist die Open-Source-Lizenz für den Datensatz, der von Xiang Zhang et al. veröffentlicht wurde. Die Datei `amazon-reviews.csv` ist der Datensatz, den Sie im Tutorial analysieren.

## Erstellen eines Amazon-S3-Buckets

Nachdem Sie den Beispieldatensatz heruntergeladen haben, erstellen Sie einen Amazon S3 S3-Bucket, um Ihre Eingabe- und Ausgabedaten zu speichern. Sie können einen S3-Bucket mit der Amazon S3 S3-Konsole oder der AWS Command Line Interface (AWS CLI) erstellen.

### Erstellen Sie einen Amazon S3 S3-Bucket (Konsole)

In der Amazon S3 S3-Konsole erstellen Sie einen Bucket mit einem Namen, der in allen Fällen eindeutig ist AWS.

### Um einen S3-Bucket (Konsole) zu erstellen

1. Melden Sie sich bei der Amazon S3 S3-Konsole an AWS Management Console und öffnen Sie sie unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie unter Buckets die Option Create Bucket aus.
3. Geben Sie unter Bucket-Name einen weltweit eindeutigen Namen ein, der den Zweck des Buckets beschreibt.
4. Wählen Sie unter Region die AWS Region aus, in der Sie den Bucket erstellen möchten. Die von Ihnen gewählte Region muss Amazon Comprehend unterstützen. Um die Latenz zu reduzieren, wählen Sie die AWS Region aus, die Ihrem geografischen Standort am nächsten liegt und von Amazon Comprehend unterstützt wird. Eine Liste der Regionen, die Amazon Comprehend unterstützen, finden Sie in der [Regionentabelle](#) im Global Infrastructure Guide.
5. Behalten Sie die Standardeinstellungen für Objekteigentum, Bucket-Einstellungen für Block Public Access, Bucket-Versionierung und Tags bei.
6. Wählen Sie für Standardverschlüsselung die Option Deaktivieren aus.

#### Tip

In diesem Tutorial wird zwar keine Verschlüsselung verwendet, Sie sollten jedoch bei der Analyse wichtiger Daten Verschlüsselung verwenden. Zur end-to-end Verschlüsselung können Sie Ihre Daten verschlüsseln, wenn sie sich im Bucket befinden, und auch, wenn

Sie Analysejobs ausführen. Weitere Informationen zur Verschlüsselung mit finden Sie AWS unter [Was ist AWS Key Management Service?](#) im AWS Key Management Service Entwicklerhandbuch.

7. Überprüfen Sie Ihre Bucket-Konfigurationen und wählen Sie dann Create Bucket aus.

Erstellen Sie einen Amazon S3 S3-Bucket (AWS CLI)

Nach dem Öffnen von führen Sie den `create-bucket` Befehl aus AWS CLI, um einen Bucket zu erstellen, in dem die Eingabe- und Ausgabedaten gespeichert werden.

Um einen Amazon S3 S3-Bucket zu erstellen (AWS CLI)

1. Um Ihren Bucket zu erstellen, führen Sie den folgenden Befehl in der aus AWS CLI. Ersetzen Sie `DOC-EXAMPLE-BUCKET` durch einen Namen für den Bucket, der in allen Fällen eindeutig ist.  
AWS

```
aws s3api create-bucket --bucket DOC-EXAMPLE-BUCKET
```

Standardmäßig erstellt der `create-bucket` Befehl einen Bucket in der Region. `us-east-1` AWS Um einen Bucket in einem AWS-Region anderen als zu erstellen `us-east-1`, fügen Sie den `LocationConstraint` Parameter zur Angabe Ihrer Region hinzu. Mit dem folgenden Befehl wird beispielsweise ein Bucket in der `us-west-2` Region erstellt.

```
aws s3api create-bucket --bucket DOC-EXAMPLE-BUCKET  
--region us-west-2 --create-bucket-configuration LocationConstraint=us-west-2
```

Beachten Sie, dass Amazon Comprehend nur in bestimmten Regionen unterstützt wird. Eine Liste der Regionen, die Amazon Comprehend unterstützen, finden Sie in der [Regionentabelle](#) im Global Infrastructure Guide.

2. Führen Sie den folgenden Befehl aus, um sicherzustellen, dass Ihr Bucket erfolgreich erstellt wurde. Der Befehl listet alle S3-Buckets auf, die Ihrem Konto zugeordnet sind.

```
aws s3 ls
```

## (nur Konsole) Ordner erstellen

Erstellen Sie als Nächstes zwei Ordner in Ihrem S3-Bucket. Der erste Ordner ist für Ihre Eingabedaten. Im zweiten Ordner sendet Amazon Comprehend die Analyseergebnisse. Wenn Sie die Amazon S3 S3-Konsole verwenden, müssen Sie die Ordner manuell erstellen. Wenn Sie die verwenden AWS CLI, können Sie Ordner erstellen, wenn Sie den Beispieldatensatz hochladen oder einen Analysejob ausführen. Aus diesem Grund bieten wir ein Verfahren zum Erstellen von Ordnern nur für Konsolenbenutzer an. Wenn Sie die AWS CLI verwenden, erstellen Sie Ordner in [Laden Sie die Eingabedaten hoch](#) und in [Schritt 3: Ausführen von Analyseaufträgen für Dokumente in Amazon S3](#).

Um Ordner in Ihrem S3-Bucket (Konsole) zu erstellen

1. Öffnen Sie die Amazon-S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie unter Buckets Ihren Bucket aus der Buckets-Liste aus.
3. Wählen Sie auf der Registerkarte Übersicht die Option Ordner erstellen aus.
4. Geben Sie für den neuen Ordnernamen ein `input`.
5. Wählen Sie für die Verschlüsselungseinstellungen „Keine“ (Bucket-Einstellungen verwenden).
6. Wählen Sie Speichern.
7. Wiederholen Sie die Schritte 3 bis 6, um einen weiteren Ordner für die Ausgabe der Analysejobs zu erstellen. Geben Sie in Schritt 4 jedoch den neuen Ordnernamen ein `output`.

## Laden Sie die Eingabedaten hoch

Nachdem Sie einen Bucket erstellt haben, laden Sie den Beispieldatensatz hoch `amazon-reviews.csv`. Sie können Daten mit der Amazon S3-Konsole oder dem in S3-Buckets hochladen. AWS CLI

Laden Sie Beispieldokumente in einen Bucket (Konsole) hoch

Laden Sie in der Amazon S3 S3-Konsole die Beispieldatensatzdatei in den Eingabeordner hoch.

Um die Beispieldokumente hochzuladen (Konsole)

1. Öffnen Sie die Amazon-S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie unter Buckets Ihren Bucket aus der Buckets-Liste aus.
3. Wählen Sie den **input** Ordner aus und klicken Sie dann auf Hochladen.

4. Wählen Sie Dateien hinzufügen und wählen Sie dann die `amazon-reviews.csv` Datei auf Ihrem Computer aus.
5. Übernehmen Sie für die anderen Einstellungen die eingestellten Standardwerte.
6. Klicken Sie auf Hochladen.

Laden Sie Beispieldokumente in einen Bucket hoch (AWS CLI)

Erstellen Sie einen Eingabeordner in Ihrem S3-Bucket und laden Sie die Datensatzdatei mit dem `cp` Befehl in den neuen Ordner hoch.

Um die Beispieldokumente hochzuladen (AWS CLI)

1. Führen Sie den folgenden AWS CLI Befehl aus, um die `amazon-reviews.csv` Datei in einen neuen Ordner in Ihrem Bucket hochzuladen. Ersetzen Sie `DOC-EXAMPLE-BUCKET` durch den Namen Ihres Buckets. Durch Hinzufügen des Pfads `/input/` am Ende erstellt Amazon S3 automatisch einen neuen Ordner namens `input` in Ihrem Bucket und lädt die Datensatzdatei in diesen Ordner hoch.

```
aws s3 cp amazon-reviews.csv s3://DOC-EXAMPLE-BUCKET/input/
```

2. Um sicherzustellen, dass Ihre Datei erfolgreich hochgeladen wurde, führen Sie den folgenden Befehl aus. Der Befehl listet den Inhalt des `input` Ordners Ihres Buckets auf.

```
aws s3 ls s3://DOC-EXAMPLE-BUCKET/input/
```

Jetzt haben Sie einen S3-Bucket mit der `amazon-reviews.csv` Datei in einem Ordner namens `input`. Wenn Sie die Konsole verwendet haben, haben Sie auch einen `output` Ordner im Bucket. Wenn Sie den verwendet haben AWS CLI, erstellen Sie den Ausgabeordner, wenn Sie die Amazon Comprehend Comprehend-Analysejobs ausführen.

## Schritt 2: (nur CLI) Erstellen einer IAM-Rolle für Amazon Comprehend

Dieser Schritt ist nur erforderlich, wenn Sie das AWS Command Line Interface (AWS CLI) verwenden, um dieses Tutorial abzuschließen. Wenn Sie die Amazon Comprehend Comprehend-Konsole zum Ausführen der Analyseaufträge verwenden, fahren Sie mit fort. [Schritt 3: Ausführen von Analyseaufträgen für Dokumente in Amazon S3](#)



Um Analyseaufträge auszuführen, benötigt Amazon Comprehend Zugriff auf den Amazon S3 S3-Bucket, der den Beispieldatensatz und die Ausgabe der Jobs enthält. Mit IAM-Rollen können Sie die Berechtigungen von AWS Diensten oder Benutzern steuern. In diesem Schritt erstellen Sie eine IAM-Rolle für Amazon Comprehend. Anschließend erstellen Sie eine ressourcenbasierte Richtlinie, die Amazon Comprehend Zugriff auf Ihren S3-Bucket gewährt, und fügen sie dieser Rolle hinzu. Am Ende dieses Schritts verfügt Amazon Comprehend über die erforderlichen Berechtigungen, um auf Ihre Eingabedaten zuzugreifen, Ihre Ausgabe zu speichern und Stimmungs- und Entitätsanalyseaufträge auszuführen.

Weitere Informationen zur Verwendung von IAM mit Amazon Comprehend finden Sie unter. [So funktioniert Amazon Comprehend mit IAM](#)

## Themen

- [Voraussetzungen](#)
- [Erstellen einer IAM-Rolle](#)
- [Fügen Sie der IAM-Rolle eine IAM-Richtlinie hinzu](#)

## Voraussetzungen

Bevor Sie beginnen, führen Sie die folgenden Schritte aus:

- Schließen Sie [Schritt 1: Dokumente zu Amazon S3 hinzufügen](#) ab.
- Verwenden Sie einen Code- oder Texteditor, um JSON-Richtlinien zu speichern und den Überblick über Ihre Amazon-Ressourcennamen (ARNs) zu behalten.

## Erstellen einer IAM-Rolle

Um auf Ihren Amazon Simple Storage Service (Amazon S3) -Bucket zugreifen zu können, muss Amazon Comprehend eine AWS Identity and Access Management (IAM) -Rolle übernehmen. Die IAM-Rolle erklärt Amazon Comprehend als vertrauenswürdige Entität. Nachdem Amazon Comprehend die Rolle übernommen hat und zu einer vertrauenswürdigen Entität geworden ist, können Sie Amazon Comprehend Bucket-Zugriffsberechtigungen gewähren. In diesem Schritt erstellen Sie eine Rolle, die Amazon Comprehend als vertrauenswürdige Entität kennzeichnet. Sie können eine Rolle mit der AWS CLI oder der Amazon Comprehend Comprehend-Konsole erstellen. Um die Konsole zu verwenden, springen Sie zu. [Schritt 3: Ausführen von Analyseaufträgen für Dokumente in Amazon S3](#)

In der Amazon Comprehend Comprehend-Konsole können Sie Rollen auswählen, bei denen der Rollename „Comprehend“ enthält und die Vertrauensrichtlinie Folgendes umfasst.

comprehend.amazonaws.com Konfigurieren Sie Ihre von der CLI erstellten Rollen so, dass sie diese Kriterien erfüllen, wenn sie in der Konsole angezeigt werden sollen.

So erstellen Sie eine IAM-Rolle für Amazon Comprehend (CLI)AWS

1. Speichern Sie die folgende Vertrauensrichtlinie als JSON-Dokument, das `comprehend-trust-policy.json` in einem Code- oder Texteditor auf Ihrem Computer aufgerufen wird. Diese Vertrauensrichtlinie erklärt Amazon Comprehend als vertrauenswürdige Entität und ermöglicht es ihm, eine IAM-Rolle zu übernehmen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "comprehend.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

2. Führen Sie den folgenden Befehl aus, um die IAM-Rolle zu erstellen. AWS CLI Der Befehl erstellt eine IAM-Rolle namens `AmazonComprehendServiceRole-access-role` und fügt der Rolle die Vertrauensrichtlinie hinzu. *path*/Ersetzen Sie es durch den Pfad Ihres lokalen Computers zum JSON-Dokument.

```
aws iam create-role --role-name AmazonComprehendServiceRole-access-role
--assume-role-policy-document file://path/comprehend-trust-policy.json
```

#### Tip

Wenn Sie die Meldung „Fehler beim Parsen des Parameters“ erhalten, ist der Pfad zu Ihrer JSON-Vertrauensrichtliniendatei wahrscheinlich falsch. Geben Sie den relativen Pfad zu der Datei an, der auf Ihrem Home-Verzeichnis basiert.

3. Kopieren Sie den Amazon-Ressourcennamen (ARN) und speichern Sie ihn in einem Texteditor. Der ARN hat ein ähnliches Format wie `arn:aws:iam::123456789012:role/AmazonComprehendServiceRole-access-role`. Sie benötigen diesen ARN, um Amazon Comprehend Comprehend-Analysejobs auszuführen.

## Fügen Sie der IAM-Rolle eine IAM-Richtlinie hinzu

Um auf Ihren Amazon S3 S3-Bucket zugreifen zu können, benötigt Amazon Comprehend Berechtigungen zum Auflisten, Lesen und Schreiben. Um Amazon Comprehend die erforderlichen Berechtigungen zu erteilen, erstellen Sie eine IAM-Richtlinie und fügen Sie sie Ihrer IAM-Rolle hinzu. Die IAM-Richtlinie ermöglicht es Amazon Comprehend, die Eingabedaten aus Ihrem Bucket abzurufen und Analyseergebnisse in den Bucket zu schreiben. Nachdem Sie die Richtlinie erstellt haben, fügen Sie sie Ihrer IAM-Rolle hinzu.

Um eine IAM-Richtlinie zu erstellen (AWS CLI)

1. Speichern Sie die folgende Richtlinie lokal als JSON-Dokument mit dem Namen `comprehend-access-policy.json`. Es gewährt Amazon Comprehend Zugriff auf den angegebenen S3-Bucket.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
      ],
      "Effect": "Allow"
    },
    {
      "Action": [
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::DOC-EXAMPLE-BUCKET"
      ],
      "Effect": "Allow"
    }
  ],
}
```

```
{
  "Action": [
    "s3:PutObject"
  ],
  "Resource": [
    "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
  ],
  "Effect": "Allow"
}
]
```

2. Führen Sie den folgenden AWS CLI Befehl aus, um die S3-Bucket-Zugriffsrichtlinie zu erstellen. *path*/Ersetzen Sie es durch den Pfad Ihres lokalen Computers zum JSON-Dokument.

```
aws iam create-policy --policy-name comprehend-access-policy
--policy-document file://path/comprehend-access-policy.json
```

3. Kopieren Sie den ARN für die Zugriffsrichtlinie und speichern Sie ihn in einem Texteditor. Der ARN hat ein ähnliches Format wie *arn:aws:iam::123456789012:policy/comprehend-access-policy*. Sie benötigen diesen ARN, um Ihre Zugriffsrichtlinie an Ihre IAM-Rolle anzuhängen.

Um die IAM-Richtlinie an Ihre IAM-Rolle anzuhängen ( )AWS CLI

- Führen Sie den folgenden Befehl aus. *policy-arn* Ersetzen Sie es durch den ARN für die Zugriffsrichtlinie, den Sie im vorherigen Schritt kopiert haben.

```
aws iam attach-role-policy --policy-arn policy-arn
--role-name AmazonComprehendServiceRole-access-role
```

Sie haben jetzt eine IAM-Rolle namens `AmazonComprehendServiceRole-access-role`, die eine Vertrauensrichtlinie für Amazon Comprehend und eine Zugriffsrichtlinie hat, die Amazon Comprehend Zugriff auf Ihren S3-Bucket gewährt. Sie haben auch den ARN für die IAM-Rolle in einen Texteditor kopiert.

## Schritt 3: Ausführen von Analyseaufträgen für Dokumente in Amazon S3

Nachdem Sie die Daten in Amazon S3 gespeichert haben, können Sie mit der Ausführung von Amazon Comprehend Comprehend-Analysejobs beginnen. Ein Stimmungsanalyseauftrag bestimmt die allgemeine Stimmung eines Dokuments (positiv, negativ, neutral oder gemischt). Ein Auftrag zur Entitätsanalyse extrahiert die Namen realer Objekte aus einem Dokument. Zu diesen Objekten gehören Personen, Orte, Titel, Ereignisse, Daten, Mengen, Produkte und Organisationen. In diesem Schritt führen Sie zwei Amazon Comprehend Comprehend-Analysejobs aus, um die Stimmung und die Entitäten aus dem Beispieldatensatz zu extrahieren.

### Themen

- [Voraussetzungen](#)
- [Analysieren Sie Stimmungen und Entitäten](#)

### Voraussetzungen

Bevor Sie beginnen, führen Sie die folgenden Schritte aus:

- Schließen Sie [Schritt 1: Dokumente zu Amazon S3 hinzufügen](#) ab.
- (Optional) Wenn Sie den verwenden AWS CLI, füllen Sie den ARN für Ihre IAM-Rolle aus [Schritt 2: \(nur CLI\) Erstellen einer IAM-Rolle für Amazon Comprehend](#) und halten Sie ihn bereit.

### Analysieren Sie Stimmungen und Entitäten

Der erste Job, den Sie ausführen, analysiert die Stimmung jeder Kundenbewertung im Beispieldatensatz. Der zweite Job extrahiert die Entitäten in jeder Kundenbewertung. Sie können Amazon Comprehend Comprehend-Analyseaufträge entweder mit der Amazon Comprehend Comprehend-Konsole oder mit dem ausführen. AWS CLI

#### Tip

Stellen Sie sicher, dass Sie sich in einer AWS Region befinden, die Amazon Comprehend unterstützt. Weitere Informationen finden Sie in der [Regionstabelle](#) im Global Infrastructure Guide.

## Analysieren Sie Stimmungen und Entitäten (Konsole)

Wenn Sie die Amazon Comprehend Comprehend-Konsole verwenden, erstellen Sie jeweils einen Job. Sie müssen die folgenden Schritte wiederholen, um sowohl eine Stimmungs- als auch eine Entitätsanalyse auszuführen. Beachten Sie, dass Sie für den ersten Job eine IAM-Rolle erstellen, für den zweiten Job jedoch die IAM-Rolle des ersten Jobs wiederverwenden können. Sie können die IAM-Rolle wiederverwenden, solange Sie denselben S3-Bucket und dieselben Ordner verwenden.

### Um Aufgaben zur Stimmungs- und Entitätsanalyse auszuführen (Konsole)

1. Stellen Sie sicher, dass Sie sich in derselben Region befinden, in der Sie Ihren Amazon Simple Storage Service (Amazon S3) -Bucket erstellt haben. Wenn Sie sich in einer anderen Region befinden, wählen Sie in der Navigationsleiste aus der AWS Regionsauswahl die Region aus, in der Sie Ihren S3-Bucket erstellt haben.
2. [Öffnen Sie die Amazon Comprehend Comprehend-Konsole unter https://console.aws.amazon.com/comprehend/](https://console.aws.amazon.com/comprehend/)
3. Wählen Sie Amazon Comprehend starten.
4. Wählen Sie im Navigationsbereich Analysis jobs aus.
5. Wählen Sie Job erstellen aus.
6. Gehen Sie im Abschnitt Auftragseinstellungen wie folgt vor:
  - a. Geben Sie unter Name `reviews-sentiment-analysis` ein.
  - b. Wählen Sie als Analysetyp die Option Stimmung aus.
  - c. Wählen Sie als Sprache Englisch aus.
  - d. Lassen Sie die Einstellung Jobverschlüsselung deaktiviert.
7. Gehen Sie im Bereich Eingabedaten wie folgt vor:
  - a. Wählen Sie als Datenquelle die Option Meine Dokumente aus.
  - b. Wählen Sie für S3-Standort die Option Browse S3 und wählen Sie dann Ihren Bucket aus der Buckets-Liste aus.
  - c. Wählen Sie in Ihrem S3-Bucket für Objekte Ihren `input` Ordner aus.
  - d. Wählen Sie im `input` Ordner den Beispieldatensatz aus `amazon-reviews.csv` und klicken Sie dann auf Auswählen.
  - e. Wählen Sie als Eingabeformat die Option Ein Dokument pro Zeile aus.
8. Gehen Sie im Abschnitt Ausgabedaten wie folgt vor:

- a. Wählen Sie unter S3-Standort die Option Browse S3 und wählen Sie dann Ihren Bucket aus der Buckets-Liste aus.
  - b. Wählen Sie in Ihrem S3-Bucket für Objekte den output Ordner aus und wählen Sie dann Wählen aus.
  - c. Lassen Sie die Verschlüsselung ausgeschaltet.
9. Gehen Sie im Abschnitt Zugriffsberechtigungen wie folgt vor:
- a. Wählen Sie für die IAM-Rolle die Option IAM-Rolle erstellen aus.
  - b. Wählen Sie für Zugriffsberechtigungen die Option Eingabe- und Ausgabe-S3-Buckets aus.
  - c. Geben Sie als Namenssuffix ein. `comprehend-access-role` Diese Rolle bietet Zugriff auf Ihren Amazon S3 S3-Bucket.
10. Wählen Sie Job erstellen aus.
11. Wiederholen Sie die Schritte 1—10, um einen Auftrag zur Entitätsanalyse zu erstellen. Nehmen Sie die folgenden Änderungen vor:
- a. Geben Sie in den Auftragseinstellungen für Name den Wert `einreviews-entities-analysis`.
  - b. Wählen Sie in den Jobeinstellungen für Analysetyp die Option Entitäten aus.
  - c. Wählen Sie unter Zugriffsberechtigungen die Option Bestehende IAM-Rolle verwenden aus. Wählen Sie als Rollenname die Option `AmazonComprehendServiceRole-comprehend-access-role` (dies ist dieselbe Rolle, die Sie für den Sentiment-Job erstellt haben) aus.

## Analysieren Sie Stimmungen und Entitäten ( )AWS CLI

Sie verwenden die `start-entities-detection-job` Befehle `start-sentiment-detection-job` und die, um Aufgaben zur Stimmungs- und Entitätsanalyse auszuführen. Nachdem Sie jeden Befehl ausgeführt haben, wird ein JSON-Objekt mit einem `JobId` Wert AWS CLI angezeigt, mit dem Sie auf Details zum Job zugreifen können, einschließlich des S3-Ausgabeorts.

So führen Sie Stimmungs- und Entitätsanalysejobs (AWS CLI) aus

1. Starten Sie einen Stimmungsanalyse-Job, indem Sie den folgenden Befehl in der ausführen.  
AWS CLI `arn:aws:iam::123456789012:role/comprehend-access-role` Ersetzen Sie es durch den ARN der IAM-Rolle, den Sie zuvor in einen Texteditor kopiert haben. Wenn sich Ihre AWS CLI Standardregion von der Region unterscheidet, in der Sie Ihren Amazon S3 S3-Bucket

erstellt haben, geben Sie den `--region` Parameter an und `us-east-1` ersetzen Sie ihn durch die Region, in der sich Ihr Bucket befindet.

```
aws comprehend start-sentiment-detection-job
--input-data-config S3Uri=s3://DOC-EXAMPLE-BUCKET/input/
--output-data-config S3Uri=s3://DOC-EXAMPLE-BUCKET/output/
--data-access-role-arn arn:aws:iam::123456789012:role/comprehend-access-role
--job-name reviews-sentiment-analysis
--language-code en
[--region us-east-1]
```

2. Nachdem Sie den Job eingereicht haben, kopieren Sie ihn JobId und speichern Sie ihn in einem Texteditor. Sie benötigen die JobId, um die Ausgabedateien des Analysejobs zu finden.
3. Starten Sie einen Entitätsanalysejob, indem Sie den folgenden Befehl ausführen.

```
aws comprehend start-entities-detection-job
--input-data-config S3Uri=s3://DOC-EXAMPLE-BUCKET/input/
--output-data-config S3Uri=s3://DOC-EXAMPLE-BUCKET/output/
--data-access-role-arn arn:aws:iam::123456789012:role/comprehend-access-role
--job-name reviews-entities-analysis
--language-code en
[--region us-east-1]
```

4. Nachdem Sie den Job gesendet haben, kopieren Sie ihn JobId und speichern Sie ihn in einem Texteditor.
5. Überprüfen Sie den Status Ihrer Jobs. Sie können den Fortschritt eines Jobs verfolgen, indem Sie ihn verfolgenJobId.

Führen Sie den folgenden Befehl aus, um den Fortschritt Ihres Stimmungsanalysejobs zu verfolgen. `sentiment-job-id` Ersetzen Sie ihn durch den JobId, den Sie nach dem Ausführen Ihrer Stimmungsanalyse kopiert haben.

```
aws comprehend describe-sentiment-detection-job
--job-id sentiment-job-id
```

Führen Sie den folgenden Befehl aus, um Ihren Entitätsanalysejob zu verfolgen. `entities-job-id` Ersetzen Sie ihn durch den JobId, den Sie nach dem Ausführen Ihrer Entitätsanalyse kopiert haben.

```
aws comprehend describe-entities-detection-job
```



```
--job-id entities-job-id
```

Es dauert einige Minuten `JobStatus`, bis das angezeigt wird `COMPLETED`.

Sie haben die Aufgaben zur Stimmungs- und Entitätsanalyse abgeschlossen. Beide Jobs sollten abgeschlossen sein, bevor Sie mit dem nächsten Schritt fortfahren. Es kann mehrere Minuten dauern, bis die Jobs abgeschlossen sind.

## Schritt 4: Vorbereiten der Amazon Comprehend für die Datenvisualisierung

Um die Ergebnisse der Stimmungs- und Entitätsanalyse-Jobs für die Erstellung von Datenvisualisierungen vorzubereiten, verwenden Sie `aws glue` `aws athena`. In diesem Schritt extrahieren Sie die Amazon Comprehend `Comprehend`-Ergebnisdateien. Anschließend erstellen Sie einen `AWS Glue Crawler`, der Ihre Daten untersucht und sie automatisch in Tabellen im `katalogisiert`. `AWS Glue Data Catalog` Danach greifen Sie mit `Amazon Athena` einem serverlosen und interaktiven Abfragedienst auf diese Tabellen zu und transformieren sie. Wenn Sie diesen Schritt abgeschlossen haben, sind Ihre `Amazon Comprehend Comprehend`-Ergebnisse sauber und bereit für die Visualisierung.

Bei einem Job zur Erkennung von PII-Entitäten ist die `Ausgabedatei` Klartext und kein komprimiertes Archiv. Der Name der `Ausgabedatei` entspricht dem Namen der `Eingabedatei`, wird jedoch am `.out` Ende angehängt. Sie müssen die `Ausgabedatei` nicht extrahieren. Überspringen Sie, um [die Daten in eine AWS Glue Data Catalog zu laden](#).

### Themen

- [Voraussetzungen](#)
- [Laden Sie die Ausgabe herunter](#)
- [Extrahieren Sie die Ausgabedateien](#)
- [Laden Sie die extrahierten Dateien hoch](#)
- [Laden Sie die Daten in eine AWS Glue Data Catalog](#)
- [Bereiten Sie die Daten für die Analyse vor](#)

### Voraussetzungen

Bevor Sie beginnen, führen Sie [Schritt 3: Ausführen von Analyseaufträgen für Dokumente in Amazon S3](#) durch.

## Laden Sie die Ausgabe herunter

Amazon Comprehend verwendet die Gzip-Komprimierung, um Ausgabedateien zu komprimieren und als Tar-Archiv zu speichern. Der einfachste Weg, die Ausgabedateien zu extrahieren, besteht darin, die `output.tar.gz` Archive lokal herunterzuladen.

In diesem Schritt laden Sie die Ausgabearchive für Stimmung und Entitäten herunter.

### Laden Sie die Ausgabedateien herunter (Konsole)

Um die Ausgabedateien für jeden Job zu finden, kehren Sie in der Amazon Comprehend-Konsole zum Analysejob zurück. Der Analysejob stellt den S3-Speicherort für die Ausgabe bereit, von dem Sie die Ausgabedatei herunterladen können.

### Um die Ausgabedateien herunterzuladen (Konsole)

1. Kehren Sie in der [Amazon Comprehend Comprehend-Konsole](#) im Navigationsbereich zu Analysis-Jobs zurück.
2. Wählen Sie Ihren Job zur Stimmungsanalyse. `reviews-sentiment-analysis`
3. Wählen Sie unter Ausgabe den Link aus, der neben Speicherort der Ausgabedaten angezeigt wird. Dadurch werden Sie zum `output.tar.gz` Archiv in Ihrem S3-Bucket weitergeleitet.
4. Wählen Sie auf der Registerkarte „Übersicht“ die Option Herunterladen aus.
5. Benennen Sie das Archiv auf Ihrem Computer in `umsentiment-output.tar.gz`. Da alle Ausgabedateien denselben Namen haben, können Sie auf diese Weise den Überblick über die Stimmungs- und Entitätsdateien behalten.
6. Wiederholen Sie die Schritte 1–4, um die Ausgabe Ihres `reviews-entities-analysis` Jobs zu finden und herunterzuladen. Benennen Sie das Archiv auf Ihrem Computer um `inentities-output.tar.gz`.

### Laden Sie die Ausgabedateien herunter (AWS CLI)

Um die Ausgabedateien für jeden Job zu finden, verwenden Sie den Job `JobId` Aus der Analyse, um den S3-Speicherort der Ausgabe zu ermitteln. Verwenden Sie dann den `cp` Befehl, um die Ausgabedatei auf Ihren Computer herunterzuladen.

### Um die Ausgabedateien herunterzuladen (AWS CLI)

1. Führen Sie den folgenden Befehl aus, um Details zu Ihrem Stimmungsanalyse-Job aufzulisten. Ersetzen Sie es `sentiment-job-id` durch das `SentimentJobId`, das Sie gespeichert haben.

```
aws comprehend describe-sentiment-detection-job --job-id sentiment-job-id
```

Wenn Sie den Überblick über Ihre verloren haben `JobId`, können Sie den folgenden Befehl ausführen, um alle Ihre Stimmungsjobs aufzulisten und nach Ihrem Job nach Namen zu filtern.

```
aws comprehend list-sentiment-detection-jobs  
--filter JobName="reviews-sentiment-analysis"
```

- Suchen Sie im `OutputDataConfig` Objekt nach dem `S3Uri` Wert. Der `S3Uri` Wert sollte dem folgenden Format ähneln: `s3://DOC-EXAMPLE-BUCKET/.../output/output.tar.gz`. Kopieren Sie diesen Wert in einen Texteditor.
- Führen Sie den folgenden Befehl aus, um das Sentiment-Ausgabearchiv in Ihr lokales Verzeichnis herunterzuladen. Ersetzen Sie den S3-Bucket-Pfad durch den Pfad, den `S3Uri` Sie im vorherigen Schritt kopiert haben. `path/` Ersetzen Sie ihn durch den Ordnerpfad zu Ihrem lokalen Verzeichnis. Der Name `sentiment-output.tar.gz` ersetzt den ursprünglichen Archivnamen, sodass Sie den Überblick über die Stimmungs- und Entitätsdateien behalten können.

```
aws s3 cp s3://DOC-EXAMPLE-BUCKET/.../output/output.tar.gz  
path/sentiment-output.tar.gz
```

- Führen Sie den folgenden Befehl aus, um Details zu Ihrem Entitätsanalysejob aufzulisten.

```
aws comprehend describe-entities-detection-job  
--job-id entities-job-id
```

Wenn Sie Ihren Job nicht kennen `JobId`, führen Sie den folgenden Befehl aus, um alle Ihre Entitäts-Jobs aufzulisten und nach Ihrem Job nach Namen zu filtern.

```
aws comprehend list-entities-detection-jobs  
--filter JobName="reviews-entities-analysis"
```

- Kopieren Sie den `S3Uri` Wert aus dem `OutputDataConfig` Objekt in der Stellenbeschreibung Ihrer Entitäten.
- Führen Sie den folgenden Befehl aus, um das Ausgabearchiv der Entitäten in Ihr lokales Verzeichnis herunterzuladen. Ersetzen Sie den S3-Bucket-Pfad durch den Pfad, den `S3Uri` Sie im vorherigen Schritt kopiert haben. `path/` Ersetzen Sie ihn durch den Ordnerpfad zu

Ihrem lokalen Verzeichnis. Der Name `entities-output.tar.gz` ersetzt den ursprünglichen Archivnamen.

```
aws s3 cp s3://DOC-EXAMPLE-BUCKET/.../output/output.tar.gz  
path/entities-output.tar.gz
```

## Extrahieren Sie die Ausgabedateien

Bevor Sie auf die Ergebnisse von Amazon Comprehend zugreifen können, müssen Sie die Stimmungs- und Entitätsarchive entpacken. Sie können entweder Ihr lokales Dateisystem oder ein Terminal verwenden, um die Archive zu entpacken.

### Extrahieren Sie die Ausgabedateien (GUI-Dateisystem)

Wenn Sie macOS verwenden, doppelklicken Sie in Ihrem GUI-Dateisystem auf das Archiv, um die Ausgabedatei aus dem Archiv zu extrahieren.

Wenn Sie Windows verwenden, können Sie ein Drittanbieter-Tool wie 7-Zip verwenden, um die Ausgabedateien in Ihr GUI-Dateisystem zu extrahieren. In Windows müssen Sie zwei Schritte ausführen, um auf die Ausgabedatei im Archiv zuzugreifen. Dekomprimieren Sie zuerst das Archiv und extrahieren Sie dann das Archiv.

Benennen Sie die Stimmungsdatei in `sentiment-output` und die Entitätsdatei `entities-output` um, um zwischen den Ausgabedateien zu unterscheiden.

### Extrahieren Sie die Ausgabedateien (Terminal)

Wenn Sie Linux oder macOS verwenden, können Sie Ihr Standardterminal verwenden. Wenn Sie Windows verwenden, müssen Sie Zugriff auf eine Umgebung im UNIX-Stil wie Cygwin haben, um Tar-Befehle ausführen zu können.

Um die Stimmungsausgabedatei aus dem Stimmungsarchiv zu extrahieren, führen Sie den folgenden Befehl in Ihrem lokalen Terminal aus.

```
tar -xvf sentiment-output.tar.gz --transform 's,^,sentiment-,'
```

Beachten Sie, dass der `--transform` Parameter der Ausgabedatei innerhalb des Archivs das Präfix `sentiment-` hinzufügt und die Datei umbenennt als `sentiment-output`. Auf diese Weise können Sie zwischen den Sentiment- und Entity-Ausgabedateien unterscheiden und ein Überschreiben verhindern.

Um die Ausgabedatei der Entitäten aus dem Entitätsarchiv zu extrahieren, führen Sie den folgenden Befehl in Ihrem lokalen Terminal aus.

```
tar -xvf entities-output.tar.gz --transform 's,^,entities-,'
```

Der `--transform` Parameter fügt dem Namen der Ausgabedatei das Präfix `entities-` hinzu.

### Tip

Um Speicherkosten in Amazon S3 zu sparen, können Sie die Dateien vor dem Hochladen erneut mit Gzip komprimieren. Es ist wichtig, die Originalarchive zu dekomprimieren und zu entpacken, da Daten aus einem Tar-Archiv nicht automatisch gelesen werden können. AWS Glue kann jedoch aus Dateien im Gzip-Format lesen.

## Laden Sie die extrahierten Dateien hoch

Laden Sie die Dateien nach dem Extrahieren in Ihren Bucket hoch. Sie müssen die Sentiment- und Entity-Ausgabedateien in separaten Ordnern speichern, damit die Daten AWS Glue korrekt gelesen werden können. Erstellen Sie in Ihrem Bucket einen Ordner für die extrahierten Stimmungsergebnisse und einen zweiten Ordner für die extrahierten Entitätsergebnisse. Sie können Ordner entweder mit der Amazon S3 S3-Konsole oder der erstellen AWS CLI.

### Laden Sie die extrahierten Dateien auf Amazon S3 (Konsole) hoch

Erstellen Sie in Ihrem S3-Bucket einen Ordner für die extrahierte Stimmungsergebnisdatei und einen Ordner für die Ergebnisdatei der Entitäten. Laden Sie dann die extrahierten Ergebnisdateien in die entsprechenden Ordner hoch.

### Um die extrahierten Dateien auf Amazon S3 (Konsole) hochzuladen

1. Öffnen Sie die Amazon-S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie unter Buckets Ihren Bucket aus und klicken Sie dann auf Ordner erstellen.
3. Geben Sie den neuen Ordernamen ein `sentiment-results` und wählen Sie Speichern. Dieser Ordner wird die extrahierte Stimmungsausgabedatei enthalten.
4. Wählen Sie auf der Registerkarte „Übersicht“ Ihres Buckets aus der Liste der Bucket-Inhalte den neuen Ordner `sentiment-results` aus. Klicken Sie auf Hochladen.

5. Wählen Sie Dateien hinzufügen, wählen Sie die `sentiment-output` Datei von Ihrem lokalen Computer aus und klicken Sie dann auf Weiter.
6. Behalten Sie die Optionen für „Benutzer verwalten“, „Zugriff für andere AWS-Konto“ und „Öffentliche Berechtigungen verwalten“ als Standardwerte bei. Wählen Sie Weiter aus.
7. Wählen Sie als Speicherklasse die Option Standard aus. Behalten Sie die Optionen für Verschlüsselung, Metadaten und Tag als Standardwerte bei. Wählen Sie Weiter aus.
8. Überprüfen Sie die Upload-Optionen und wählen Sie dann Upload.
9. Wiederholen Sie die Schritte 1–8, um einen Ordner mit dem Namen zu erstellen `entities-results`, und laden Sie die `entities-output` Datei in diesen Ordner hoch.

Laden Sie die extrahierten Dateien auf Amazon S3 hoch (AWS CLI)

Sie können einen Ordner in Ihrem S3-Bucket erstellen, während Sie eine Datei mit dem `cp` Befehl hochladen.

Um die extrahierten Dateien auf Amazon S3 hochzuladen (AWS CLI)

1. Erstellen Sie einen Stimmungsordner und laden Sie Ihre Stimmungsdatei in diesen hoch, indem Sie den folgenden Befehl ausführen. `path/` Ersetzen Sie ihn durch den lokalen Pfad zu Ihrer extrahierten Stimmungsausgabedatei.

```
aws s3 cp path/sentiment-output s3://DOC-EXAMPLE-BUCKET/sentiment-results/
```

2. Erstellen Sie einen Ausgabeordner für Entitäten und laden Sie Ihre Entitätsdatei in diesen Ordner hoch, indem Sie den folgenden Befehl ausführen. `path/` Ersetzen Sie ihn durch den lokalen Pfad zu Ihrer extrahierten Entitäten-Ausgabedatei.

```
aws s3 cp path/entities-output s3://DOC-EXAMPLE-BUCKET/entities-results/
```

## Laden Sie die Daten in eine AWS Glue Data Catalog

Um die Ergebnisse in eine Datenbank zu übertragen, können Sie einen AWS Glue Crawler verwenden. Ein AWS Glue Crawler scannt Dateien und erkennt das Schema der Daten.

Anschließend ordnet er die Daten in Tabellen in einer AWS Glue Data Catalog (einer serverlosen Datenbank) an. Sie können einen Crawler mit der AWS Glue Konsole oder dem erstellen. AWS CLI

## Laden Sie die Daten in eine AWS Glue Data Catalog (Konsole)

Erstellen Sie einen AWS Glue Crawler, der Ihre `entities-results` Ordner `sentiment-results` und Ordner separat scannt. Eine neue IAM-Rolle für AWS Glue gibt dem Crawler die Erlaubnis, auf Ihren S3-Bucket zuzugreifen. Sie erstellen diese IAM-Rolle beim Einrichten des Crawlers.

Um die Daten in eine AWS Glue Data Catalog (Konsole) zu laden

1. Stellen Sie sicher, dass Sie sich in einer Region befinden, die unterstützt AWS Glue. Wenn Sie sich in einer anderen Region befinden, wählen Sie in der Navigationsleiste in der Regionsauswahl eine unterstützte Region aus. Eine Liste der Regionen, die unterstützt werden AWS Glue, finden Sie in der [Regionentabelle](#) im Global Infrastructure Guide.
2. Öffnen Sie die AWS Glue Konsole unter <https://console.aws.amazon.com/glue/>.
3. Wählen Sie im Navigationsbereich Crawlers und dann Crawler hinzufügen aus.
4. Geben Sie als Crawler-Name Folgendes ein **comprehend-analysis-crawler** und wählen Sie dann Weiter aus.
5. Wählen Sie als Crawler-Quellentyp die Option Datenspeicher und dann Weiter aus.
6. Gehen Sie unter Datenspeicher hinzufügen wie folgt vor:
  - a. Wählen Sie in Choose a data store (Datenspeicher auswählen) die Option S3 aus.
  - b. Lassen Sie das Feld Verbindung leer.
  - c. Wählen Sie unter Daten durchsuchen die Option Spezifizierter Pfad in meinem Konto aus.
  - d. Geben Sie unter Pfad einschließen den vollständigen S3-Pfad des Stimmungsausgabeordners ein: `s3://DOC-EXAMPLE-BUCKET/sentiment-results`
  - e. Wählen Sie Weiter aus.
7. Wählen Sie für Anderen Datenspeicher hinzufügen die Option Ja und dann Weiter aus. Wiederholen Sie Schritt 6, geben Sie jedoch den vollständigen S3-Pfad des Entitäten-Ausgabeordners ein: `s3://DOC-EXAMPLE-BUCKET/entities-results`.
8. Wählen Sie für Anderen Datenspeicher hinzufügen die Option Nein und dann Weiter aus.
9. Gehen Sie unter Wählen Sie eine IAM-Rolle wie folgt vor:
  - a. Wählen Sie Create an IAM-Rolle aus.
  - b. Geben Sie für die IAM-Rolle den Text ein **glue-access-role** und wählen Sie dann Weiter aus.

10. Wählen Sie unter Einen Zeitplan für diesen Crawler erstellen die Option Bei Bedarf ausführen und dann Weiter aus.
11. Gehen Sie wie folgt vor, um die Ausgabe des Crawlers zu konfigurieren:
  - a. Wählen Sie für Datenbank die Option Datenbank hinzufügen aus.
  - b. Für Database name (Datenbankname) geben Sie `comprehend-results` ein. In dieser Datenbank werden Ihre Amazon Comprehend Comprehend-Ausgabetablen gespeichert.
  - c. Behalten Sie die Standardeinstellungen der anderen Optionen bei und wählen Sie Weiter.
12. Überprüfen Sie die Crawler-Informationen und wählen Sie dann Fertig stellen.
13. Wählen Sie in der Glue-Konsole unter Crawlers die Option `comprehend-analysis-crawler` Crawler ausführen aus. Es kann einige Minuten dauern, bis der Crawler fertig ist.

Laden Sie die Daten in ein AWS Glue Data Catalog ( )AWS CLI

Erstellen Sie eine IAM-Rolle für AWS Glue den Zugriff auf Ihren S3-Bucket. Erstellen Sie dann eine Datenbank in der AWS Glue Data Catalog. Erstellen Sie abschließend einen Crawler und führen Sie ihn aus, der Ihre Daten in Tabellen in der Datenbank lädt.

Um die Daten in ein AWS Glue Data Catalog ( )AWS CLI zu laden

1. Gehen Sie wie folgt vor AWS Glue, um eine IAM-Rolle für zu erstellen:
  - a. Speichern Sie die folgende Vertrauensrichtlinie als JSON-Dokument, das `glue-trust-policy.json` auf Ihrem Computer aufgerufen wird.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "glue.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```



- b. Führen Sie den folgenden Befehl aus, um eine IAM-Rolle zu erstellen. *path*/Ersetzen Sie es durch den Pfad Ihres lokalen Computers zum JSON-Dokument.

```
aws iam create-role --role-name glue-access-role
--assume-role-policy-document file://path/glue-trust-policy.json
```

- c. Wenn die Amazon-Ressourcennummer (ARN) für die neue Rolle AWS CLI aufgeführt ist, kopieren Sie sie und speichern Sie sie in einem Texteditor.
- d. Speichern Sie die folgende IAM-Richtlinie als JSON-Dokument, das `glue-access-policy.json` auf Ihrem Computer aufgerufen wird. Die Richtlinie gewährt die AWS Glue Erlaubnis, Ihre Ergebnisordner zu crawlen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::DOC-EXAMPLE-BUCKET/sentiment-results*",
        "arn:aws:s3:::DOC-EXAMPLE-BUCKET/entities-results*"
      ]
    }
  ]
}
```

- e. Führen Sie den folgenden Befehl aus, um die IAM-Richtlinie zu erstellen. *path*/Ersetzen Sie es durch den Pfad Ihres lokalen Computers zum JSON-Dokument.

```
aws iam create-policy --policy-name glue-access-policy
--policy-document file://path/glue-access-policy.json
```

- f. Wenn der den ARN der Zugriffsrichtlinie AWS CLI auflistet, kopieren Sie ihn und speichern Sie ihn in einem Texteditor.
- g. Fügen Sie die neue Richtlinie der IAM-Rolle hinzu, indem Sie den folgenden Befehl ausführen. *policy-arn* Ersetzen Sie es durch den ARN für die IAM-Richtlinie, den Sie im vorherigen Schritt kopiert haben.

```
aws iam attach-role-policy --policy-arn policy-arn
--role-name glue-access-role
```

- h. Fügen Sie die AWS verwaltete Richtlinie Ihrer IAM-Rolle `AWSGlueServiceRole` hinzu, indem Sie den folgenden Befehl ausführen.

```
aws iam attach-role-policy --policy-arn
arn:aws:iam::aws:policy/service-role/AWSGlueServiceRole
--role-name glue-access-role
```

2. Erstellen Sie eine AWS Glue Datenbank, indem Sie den folgenden Befehl ausführen.

```
aws glue create-database
--database-input Name="comprehend-results"
```

3. Erstellen Sie einen neuen AWS Glue Crawler, indem Sie den folgenden Befehl ausführen. *glue-iam-role-arn* Ersetzen Sie es durch den ARN Ihrer AWS Glue IAM-Rolle.

```
aws glue create-crawler
--name comprehend-analysis-crawler
--role glue-iam-role-arn
--targets S3Targets=[
{Path="s3://DOC-EXAMPLE-BUCKET/sentiment-results"},
{Path="s3://DOC-EXAMPLE-BUCKET/entities-results"}]
--database-name comprehend-results
```

4. Starten Sie den Crawler, indem Sie den folgenden Befehl ausführen.

```
aws glue start-crawler --name comprehend-analysis-crawler
```

Es kann einige Minuten dauern, bis der Crawler fertig ist.

## Bereiten Sie die Daten für die Analyse vor

Jetzt haben Sie eine Datenbank, die mit den Amazon Comprehend Comprehend-Ergebnissen gefüllt ist. Die Ergebnisse sind jedoch verschachtelt. Um sie zu entflechten, führen Sie einige SQL-Anweisungen aus. Amazon Athena Amazon Athena ist ein interaktiver Abfrageservice, der es einfach macht, Daten in Amazon S3 mithilfe von Standard-SQL zu analysieren. Athena ist serverlos, sodass keine Infrastruktur verwaltet werden muss, und es gibt ein pay-per-query Preismodell. In diesem

Schritt erstellen Sie neue Tabellen mit bereinigten Daten, die Sie zur Analyse und Visualisierung verwenden können. Sie verwenden die Athena-Konsole, um die Daten vorzubereiten.

Um die Daten vorzubereiten

1. Öffnen Sie die Athena-Konsole unter <https://console.aws.amazon.com/athena/>.
2. Wählen Sie im Abfrage-Editor Einstellungen und anschließend Verwalten aus.
3. Geben Sie als Speicherort der Abfrageergebnisse `s3://DOC-EXAMPLE-BUCKET/query-results/`. Dadurch wird ein neuer Ordner mit dem Namen `query-results` in Ihrem Bucket erstellt, in dem die Ausgabe der von Ihnen Amazon Athena ausgeführten Abfragen gespeichert wird. Wählen Sie Speichern.
4. Wählen Sie im Abfrage-Editor Editor aus.
5. Wählen Sie für Datenbank die AWS Glue Datenbank `aws-comprehend-results`, die Sie erstellt haben.
6. Im Abschnitt Tabellen sollten Sie zwei Tabellen mit dem Namen `sentiment_results` und `habenentities_results`. Zeigen Sie eine Vorschau der Tabellen an, um sicherzustellen, dass der Crawler die Daten geladen hat. Wählen Sie in den Optionen jeder Tabelle (die drei Punkte neben dem Tabellennamen) die Option Tabellenvorschau aus. Eine kurze Abfrage wird automatisch ausgeführt. Überprüfen Sie im Ergebnisbereich, ob die Tabellen Daten enthalten.

 Tip

Wenn die Tabellen keine Daten enthalten, versuchen Sie, die Ordner in Ihrem S3-Bucket zu überprüfen. Stellen Sie sicher, dass es einen Ordner für Entitätsergebnisse und einen Ordner für Stimmungsergebnisse gibt. Versuchen Sie dann, einen neuen AWS Glue Crawler auszuführen.

7. Um die **sentiment\_results** Tabelle zu entwirren, geben Sie die folgende Abfrage in den Abfrage-Editor ein und wählen Sie Ausführen aus.

```
CREATE TABLE sentiment_results_final AS
SELECT file, line, sentiment,
sentimentscore.mixed AS mixed,
sentimentscore.negative AS negative,
sentimentscore.neutral AS neutral,
sentimentscore.positive AS positive
FROM sentiment_results
```

8. Um mit dem Aufheben der Verschachtelung der Entitätstabelle zu beginnen, geben Sie die folgende Abfrage in den Abfrage-Editor ein und wählen Sie Ausführen aus.

```
CREATE TABLE entities_results_1 AS
SELECT file, line, nested FROM entities_results
CROSS JOIN UNNEST(entities) as t(nested)
```

9. Um die Verschachtelung der Entitätstabelle zu beenden, geben Sie die folgende Abfrage in den Abfrage-Editor ein und wählen Sie Abfrage ausführen aus.

```
CREATE TABLE entities_results_final AS
SELECT file, line,
nested.beginoffset AS beginoffset,
nested.endoffset AS endoffset,
nested.score AS score,
nested.text AS entity,
nested.type AS category
FROM entities_results_1
```

Ihre **sentiment\_results\_final** Tabelle sollte wie folgt aussehen, mit den Spalten Datei, Zeile, Stimmung, gemischt, negativ, neutral und positiv. Die Tabelle sollte einen Wert pro Zelle haben. Die Stimmungsspalte beschreibt die wahrscheinlichste allgemeine Stimmung einer bestimmten Bewertung. Die gemischten, negativen, neutralen und positiven Spalten geben Werte für jeden Stimmungstyp an.

Results							
file	line	sentiment	mixed	negative	neutral	positive	
amazon-reviews.csv	6	MIXED	0.9862896203994751	0.0015502438182011247	1.6660270921420306E-4	0.0119935879483	
amazon-reviews.csv	8	POSITIVE	0.0012987082591280341	0.01186690479516983	0.174478679895401	0.8123556375503	
amazon-reviews.csv	11	POSITIVE	6.5368581090297084E-6	0.0013866390800103545	0.007405391428619623	0.9912014007568	
amazon-reviews.csv	13	POSITIVE	4.7155481297522783E-4	0.24615342915058136	0.017713148146867752	0.7356618046760	
amazon-reviews.csv	14	POSITIVE	1.5821871784282848E-5	0.06828905642032623	0.014075091108679771	0.9176200628280	
amazon-reviews.csv	16	MIXED	0.9864791035652161	8.548551704734564E-4	1.0789262159960344E-4	0.0125581491738	
amazon-reviews.csv	20	NEGATIVE	1.1621621524682269E-4	0.9815887212753296	0.004688907880336046	0.0136061981320	
amazon-reviews.csv	21	POSITIVE	4.663573781726882E-5	0.009533549658954144	0.0015825830632820725	0.9888372421264	
amazon-reviews.csv	23	POSITIVE	1.7699007003102452E-4	0.40269607305526733	0.0018250439316034317	0.5953019261360	
amazon-reviews.csv	25	POSITIVE	1.8434448065818287E-6	1.15832663141191E-4	0.0010993879986926913	0.9987829327583	

Ihre **entities\_results\_final** Tabelle sollte wie folgt aussehen, mit Spalten mit den Namen file, line, beginoffset, endoffset, score, entity und category. Die Tabelle sollte einen Wert pro Zelle haben. Die Bewertungsspalte gibt das Vertrauen von Amazon Comprehend in die erkannte Entität an. Die Kategorie gibt an, welche Art von Entität Comprehend erkannt wurde.

Results							
file	line	beginoffset	endoffset	score	entity	category	
amazon-reviews.csv	0	15	22	0.9885989378545348	English	OTHER	
amazon-reviews.csv	2	24	28	0.9699371997593782	2 me	QUANTITY	
amazon-reviews.csv	2	94	95	0.6523066984191679	2	QUANTITY	
amazon-reviews.csv	2	125	126	0.713791396412543	2	QUANTITY	
amazon-reviews.csv	4	30	36	0.9957169942979278	kindle	COMMERCIAL_ITEM	
amazon-reviews.csv	5	1	10	0.9979111763962706	Hawthorne	PERSON	
amazon-reviews.csv	5	135	142	0.5065408081314243	Puritan	OTHER	
amazon-reviews.csv	5	143	148	0.7702269458801602	Salem	LOCATION	
amazon-reviews.csv	5	211	229	0.999675563687763	The Scarlet Letter	TITLE	
amazon-reviews.csv	5	233	236	0.8944631322676461	one	QUANTITY	

Nachdem Sie die Amazon Comprehend Comprehend-Ergebnisse in Tabellen geladen haben, können Sie die Daten visualisieren und aussagekräftige Erkenntnisse daraus gewinnen.

## Schritt 5: Visualisieren der Amazon Comprehend-Ausgabe in Amazon QuickSight

Nachdem Sie die Amazon Comprehend-Ergebnisse in Tabellen gespeichert haben, können Sie eine Verbindung zu herstellen und die Daten mit Amazon visualisieren QuickSight. Amazon QuickSight ist ein von AWS verwaltetes Business Intelligence (BI)-Tool zur Visualisierung von Daten. Amazon QuickSight macht es einfach, eine Verbindung zu Ihrer Datenquelle herzustellen und leistungsstarke Visualisierungen zu erstellen. In diesem Schritt verbinden Sie Amazon QuickSight mit Ihren Daten, erstellen Visualisierungen, die Erkenntnisse aus den Daten gewinnen, und veröffentlichen ein Dashboard mit Visualisierungen.

### Themen

- [Voraussetzungen](#)
- [Amazon QuickSight Zugriff gewähren](#)
- [Importieren der Datensätze](#)

- [Erstellen einer Stimmungsvisualisierung](#)
- [Erstellen einer Entitätsvisualisierung](#)
- [Veröffentlichen eines Dashboards](#)
- [Bereinigen](#)

## Voraussetzungen

Bevor Sie beginnen, führen Sie [Schritt 4: Vorbereiten der Amazon Comprehend für die Datenvisualisierung](#) durch.

## Amazon QuickSight Zugriff gewähren

Zum Importieren der Daten QuickSight benötigt Amazon Zugriff auf Ihren Amazon Simple Storage Service (Amazon S3)-Bucket und Ihre Amazon Athena Tabellen. Um Amazon QuickSight Zugriff auf Ihre Daten zu gewähren, müssen Sie als QuickSight Administrator angemeldet sein und Zugriff haben, um die Ressourcenberechtigungen zu bearbeiten. Wenn Sie die folgenden Schritte nicht ausführen können, überprüfen Sie die IAM-Voraussetzungen auf der Übersichtsseite [Tutorial: Analysieren von Erkenntnissen aus Kundenrezensionen mit Amazon Comprehend](#).

So gewähren Sie Amazon QuickSight Zugriff auf Ihre Daten

1. Öffnen Sie die [Amazon QuickSight-Konsole](#).
2. Wenn Sie Amazon zum ersten Mal verwenden QuickSight, werden Sie von der Konsole aufgefordert, einen neuen Administratorbenutzer zu erstellen, indem Sie eine E-Mail-Adresse angeben. Geben Sie für E-Mail-Adresse dieselbe E-Mail-Adresse wie Ihr ein AWS-Konto. Klicken Sie auf Weiter.
3. Wählen Sie nach der Anmeldung Ihren Profilnamen in der Navigationsleiste und dann Verwalten QuickSight aus. Sie müssen als Administrator angemeldet sein, um die Option Verwalten QuickSight anzeigen zu können.
4. Wählen Sie Sicherheit und Berechtigungen aus.
5. Wählen Sie QuickSight für den Zugriff auf - AWS Services Hinzufügen oder Entfernen aus.
6. Wählen Sie Amazon S3.
7. Wählen Sie unter Amazon S3-Buckets auswählen Ihren S3-Bucket sowohl für S3-Bucket- als auch für Schreibberechtigungen für die Athena-Arbeitsgruppe aus.
8. Wählen Sie Finish (Abschließen).
9. Wählen Sie Aktualisieren.

## Importieren der Datensätze

Bevor Sie Visualisierungen erstellen, müssen Sie die Stimmungs- und Entitätsdatensätze zu Amazon hinzufügen QuickSight. Sie tun dies mit der Amazon- QuickSight Konsole. Sie importieren Ihre Tabellen für nicht verschachtelte Stimmungen und nicht verschachtelte Entitäten aus Amazon Athena.

So importieren Sie Ihre Datensätze

1. Öffnen Sie die [Amazon QuickSight-Konsole](#).
2. Wählen Sie in der Navigationsleiste unter Datensätze die Option Neuer Datensatz aus.
3. Wählen Sie für Datensatz erstellen die Option Athena aus.
4. Geben Sie für Datenquellenname ein `reviews-sentiment-analysis` und wählen Sie Datenquelle erstellen aus.
5. Für Datenbank wählen Sie die `comprehend-results`-Datenbank aus.
6. Wählen Sie für Tabellen die Stimmungstabelle `sentiment_results_final` und dann Auswählen aus.
7. Wählen Sie In SPICE importieren für schnellere Analysen und wählen Sie Visualisieren aus. SPICE ist QuickSightdie In-Memory-Berechnungs-Engine von , die beim Erstellen von Visualisierungen schnellere Analysen als direkte Abfragen bietet.
8. Kehren Sie zur Amazon- QuickSight Konsole zurück und wählen Sie Datensätze aus. Wiederholen Sie die Schritte 1-7, um einen Entitätsdatensatz zu erstellen, nehmen Sie jedoch die folgenden Änderungen vor:
  - a. Geben Sie für Datenquellenname ein `reviews-entities-analysis`.
  - b. Wählen Sie für Tabellen die Entitätstabelle `entities_results_final`.

## Erstellen einer Stimmungsvisualisierung

Nachdem Sie nun auf Ihre Daten in Amazon zugreifen können QuickSight, können Sie mit der Erstellung von Visualisierungen beginnen. Sie erstellen ein Kreisdiagramm mit den Amazon Comprehend-Emotionsdaten. Das Kreisdiagramm zeigt, welcher Anteil der Bewertungen positiv, neutral, gemischt und negativ ist.

So visualisieren Sie Stimmungsdaten

1. Wählen Sie in der Amazon- QuickSight Konsole Analysen und dann Neue Analyse aus.

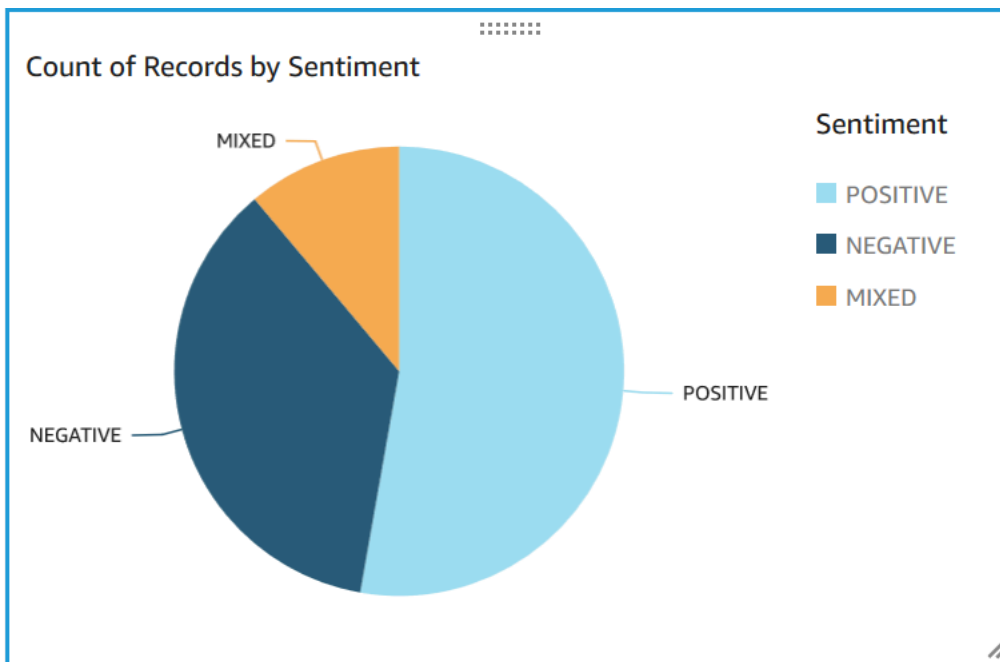
- Wählen Sie unter Ihre Datensätze den Stimmungsdatensatz `sentiment_results_final` und dann Analyse erstellen aus.
- Wählen Sie im visuellen Editor unter Feldliste die Option Stimmung aus.

**Note**

Die Werte in der Liste Felder hängen von den Spaltennamen ab, die Sie zum Erstellen der Tabellen in verwendet haben Amazon Athena. Wenn Sie die angegebenen Spaltennamen in den SQL-Abfragen geändert haben, unterscheiden sich die Feldlistennamen von den Namen, die in diesen Visualisierungsbeispielen verwendet werden.

- Wählen Sie für Visualisierungstypen die Option Kreisdiagramm aus.

Ein Kreisdiagramm ähnlich dem folgenden mit positiven, neutralen, gemischten und negativen Abschnitten wird angezeigt. Um die Anzahl und den Prozentsatz eines Abschnitts anzuzeigen, bewegen Sie den Mauszeiger darüber.



## Erstellen einer Entitätsvisualisierung

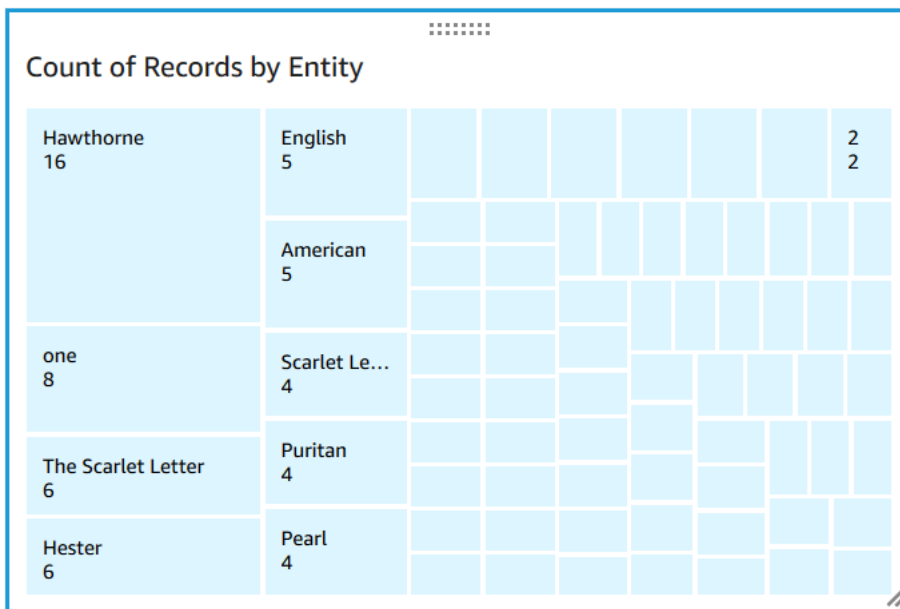
Erstellen Sie jetzt eine zweite Visualisierung mit dem Entitätsdatensatz. Sie erstellen eine Baumkarte der verschiedenen Entitäten in den Daten. Jeder Block in der Baumkarte steht für eine Entität, und die Größe des Blocks korreliert mit der Häufigkeit, mit der die Entität im Datensatz erscheint.



## So visualisieren Sie Entitätsdaten

1. Wählen Sie im Bereich Visualisierungssteuerung neben Datensatz das Symbol Datensätze hinzufügen, bearbeiten, ersetzen und entfernen aus.
2. Wählen Sie Datensatz hinzufügen.
3. Wählen Sie für Datensatz auswählen, um hinzuzufügen, Ihren Entitätsdatensatz `entities_results_final` aus der Liste der Datensätze aus und wählen Sie Auswählen aus.
4. Wählen Sie im Bereich Kontrolle visualisieren das Dropdown-Menü Datensatz und dann den Entitätsdatensatz `ausentities_results_final`.
5. Wählen Sie unter Feldliste die Entität aus.
6. Wählen Sie für Visualisierungstypen die Option Strukturzuordnung aus.

Eine Baumkarte ähnlich der folgenden wird neben Ihrem Kreisdiagramm angezeigt. Um die Anzahl einer bestimmten Entität anzuzeigen, bewegen Sie den Mauszeiger über einen Block.



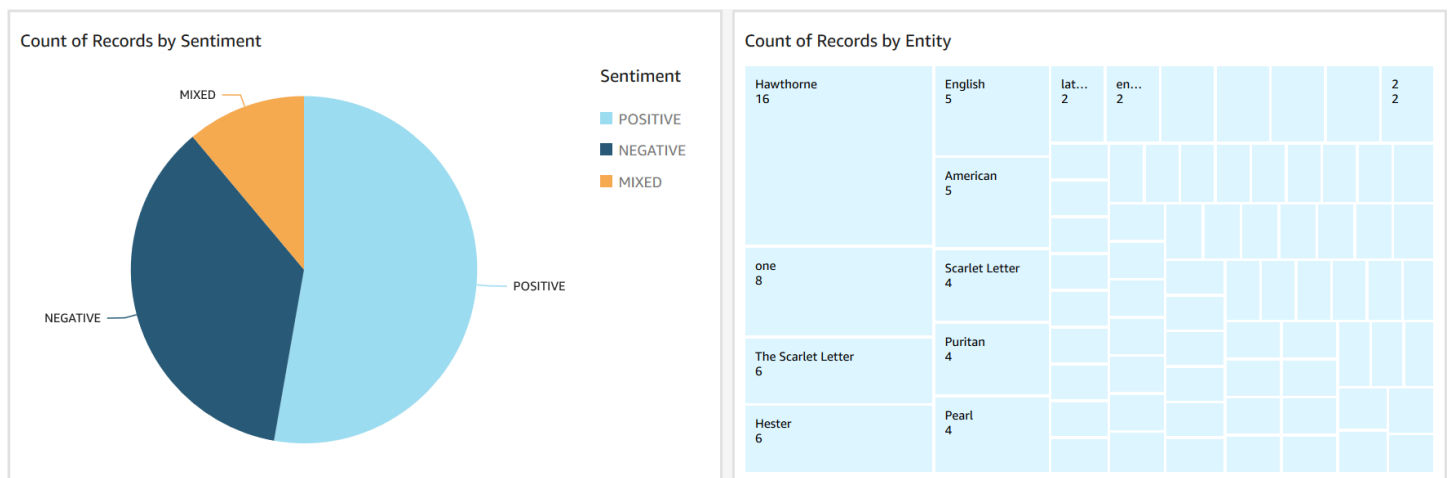
## Veröffentlichen eines Dashboards

Nachdem Sie die Visualisierungen erstellt haben, können Sie sie als Dashboard veröffentlichen. Sie können verschiedene Aufgaben mit einem Dashboard ausführen, z. B. es mit Benutzern in Ihrem teilen AWS-Konto, als PDF speichern oder als Bericht senden (beschränkt auf die Enterprise Edition von Amazon QuickSight). In diesem Schritt veröffentlichen Sie die Visualisierungen als Dashboard in Ihrem Konto.

## So veröffentlichen Sie Ihr Dashboard

1. Wählen Sie in der Navigationsleiste Freigabe aus.
2. Klicken Sie auf Publish dashboard (Dashboard veröffentlichen).
3. Wählen Sie Neues Dashboard veröffentlichen unter und geben Sie den Namen `comprehend-analysis-reviews` für das Dashboard ein.
4. Klicken Sie auf Publish dashboard (Dashboard veröffentlichen).
5. Schließen Sie den Bereich Dashboard mit Benutzern teilen, indem Sie die Schaltfläche Schließen in der oberen rechten Ecke auswählen.
6. Wählen Sie in der Amazon- QuickSight Konsole im Navigationsbereich Dashboards aus. Unter Dashboards `comprehend-analysis-reviews` sollte eine Miniaturansicht Ihres neuen Dashboards angezeigt werden. Wählen Sie das Dashboard aus, um es anzuzeigen.

Sie haben jetzt ein Dashboard mit Stimmungs- und Entitätsvisualisierungen, das dem folgenden Beispiel ähnelt.



### Tip

Wenn Sie die Visualisierungen in Ihrem Dashboard bearbeiten möchten, kehren Sie zu Analysen zurück und bearbeiten Sie die Visualisierung, die Sie aktualisieren möchten. Veröffentlichen Sie dann das Dashboard erneut entweder als neues Dashboard oder als Ersatz für das vorhandene Dashboard.

## Bereinigen

Nach Abschluss dieses Tutorials möchten Sie möglicherweise alle AWS Ressourcen bereinigen, die Sie nicht mehr verwenden möchten. Für aktive AWS Ressourcen können weiterhin Gebühren in Ihrem Konto anfallen.

Die folgenden Aktionen können dazu beitragen, laufende Gebühren zu vermeiden:

- Stornieren Sie Ihr Amazon- QuickSight Abonnement. Amazon QuickSight ist ein monatlicher Abonnementservice. Informationen zum Widerrufen Ihres Abonnements finden Sie unter [Stornieren Ihres Abonnements](#) im Amazon- QuickSight Benutzerhandbuch.
- Löschen Sie Ihren Amazon S3-Bucket. Amazon S3 berechnet Ihnen Speichergebühren. Um Ihre Amazon S3-Ressourcen zu bereinigen, löschen Sie Ihren Bucket. Informationen zum Löschen eines Buckets finden Sie unter [Wie lösche ich einen S3-Bucket?](#) im Benutzerhandbuch für Amazon Simple Storage Service. Stellen Sie sicher, dass Sie alle wichtigen Dateien speichern, bevor Sie Ihren Bucket löschen.
- Löschen Sie Ihre AWS Glue Data Catalog. Die AWS Glue Data Catalog berechnet Ihnen monatlich Speichergebühren. Sie können Ihre Datenbanken löschen, um laufende Gebühren zu vermeiden. Informationen zum Verwalten Ihrer AWS Glue Data Catalog Datenbanken finden Sie unter [Arbeiten mit Datenbanken in der AWS Glue Konsole](#) im AWS Glue Entwicklerhandbuch für . Stellen Sie sicher, dass Sie Ihre Daten exportieren, bevor Sie Datenbanken oder Tabellen löschen.

## Verwenden von Amazon S3-Objekt-Lambda-Zugriffspunkten für persönlich identifizierbare Informationen (PII)

Verwenden Sie Amazon S3 Object Lambda Access Points für persönlich identifizierbare Informationen (PII), um zu konfigurieren, wie Dokumente aus Ihrem Amazon S3-Bucket abgerufen werden. Sie können den Zugriff auf Dokumente steuern, die PII enthalten, und PII aus Dokumenten redigieren. Weitere Informationen dazu, wie Amazon Comprehend PII in Ihren Dokumenten erkennen kann, finden Sie unter [Erkennung von PII-Entitäten](#). Amazon S3 Object Lambda Access Points verwenden AWS Lambda Funktionen, um die Ausgabe einer standardmäßigen Amazon S3 GET-Anforderung automatisch zu transformieren. Weitere Informationen finden Sie unter [Transformieren von Objekten mit S3 Objekt Lambda](#) im Benutzerhandbuch für Amazon Simple Storage Service.

Wenn Sie einen Amazon S3 Object Lambda Access Point für PII erstellen, werden Dokumente mit Amazon Comprehend Lambda-Funktionen verarbeitet, um den Zugriff auf Dokumente zu steuern, die PII enthalten, und PII aus Dokumenten zu redigieren.

Wenn Sie einen Amazon S3 Object Lambda Access Point für PII erstellen, werden Dokumente mit den folgenden Amazon Comprehend Lambda-Funktionen verarbeitet:

- `ComprehendPiiAccessControlS3ObjectLambda` – Steuert den Zugriff auf Dokumente mit PII, die in Ihrem S3-Bucket gespeichert sind. Um weitere Informationen zu dieser Lambda-Funktion zu erhalten, melden Sie sich bei der an, AWS Management Console um die [ComprehendPiiAccessControlS3ObjectLambda](#)-Funktion in der anzuzeigenAWS Serverless Application Repository.
- `ComprehendPiiRedactionS3ObjectLambda` – Redigiert PII aus Dokumenten in Ihrem Amazon S3-Bucket. Um weitere Informationen zu dieser Lambda-Funktion zu erhalten, melden Sie sich bei der an, AWS Management Console um die [ComprehendPiiRedactionS3ObjectLambda](#)-Funktion in der anzuzeigenAWS Serverless Application Repository.

Informationen zum Bereitstellen von Serverless-Anwendungen über die AWS Serverless Application Repository finden Sie unter [Bereitstellen von Anwendungen](#) im AWS Entwicklerhandbuch für Serverless Application Repository.

Themen

- [Steuern des Zugriffs auf Dokumente mit persönlich identifizierbaren Informationen \(PII\)](#)
- [Redigieren von persönlich identifizierbaren Informationen \(PII\) aus Dokumenten](#)

## Steuern des Zugriffs auf Dokumente mit persönlich identifizierbaren Informationen (PII)

Sie können einen Amazon S3 Object Lambda Access Point verwenden, um den Zugriff auf Dokumente mit persönlich identifizierbaren Informationen (PII) zu steuern.

Um sicherzustellen, dass nur autorisierte Benutzer Zugriff auf Dokumente haben, die PII enthalten, die in Ihrem Amazon S3-Bucket gespeichert sind, verwenden Sie die `ComprehendPiiAccessControlS3ObjectLambda` Funktion. Diese Lambda-Funktion verwendet die [-ContainsPiiEntities](#) Operation bei der Verarbeitung einer standardmäßigen Amazon S3-GET-Anforderung für Dokumentobjekte.

Wenn Sie beispielsweise Dokumente in Ihrem S3-Bucket haben, die PII wie Kreditkartennummern oder Bankkontoinformationen enthalten, können Sie die `ComprehendPiiAccessControlS3ObjectLambda` Funktion so konfigurieren, dass diese PII-

Entitätstypen erkannt und der Zugriff auf nicht autorisierte Benutzer eingeschränkt wird. Weitere Informationen zu unterstützten PII-Entitätstypen finden Sie unter [Universelle PII-Entitätstypen](#).

Um weitere Informationen zu dieser Lambda-Funktion zu erhalten, melden Sie sich bei der an, AWS Management Console um die [ComprehendPiiAccessControlS3ObjectLambda](#)-Funktion in der anzuzeigenAWS Serverless Application Repository.

## Erstellen eines Amazon S3-Objekt-Lambda-Zugriffspunkts zur Steuerung des Zugriffs auf Dokumente

Im folgenden Beispiel wird ein Amazon S3 Object Lambda Access Point erstellt, um den Zugriff auf Dokumente zu steuern, die Sozialversicherungsnummern enthalten.

Erstellen eines Lambda-Zugriffspunkts für Amazon S3-Objekte mit der AWS Command Line Interface

Erstellen Sie eine Konfiguration des Amazon S3 Object Lambda Access Point und speichern Sie die Konfiguration in einer Datei namens `config.json`.

```
{
  "SupportingAccessPoint": "s3-default-access-point-name-arn",
  "TransformationConfigurations": [
    {
      "Actions": [
        "s3:GetObject"
      ],
      "ContentTransformation": {
        "AwsLambda": {
          "FunctionArn": "comprehend-pii-access-control-s3-object-lambda-arn",
          "FunctionPayload": "{\"pii_entities_types\": \"SSN\"}"
        }
      }
    }
  ]
}
```

Im folgenden Beispiel wird ein Amazon S3 Object Lambda Access Point basierend auf der in der `config.json` Datei definierten Konfiguration erstellt.

Das Beispiel ist für Unix, Linux und macOS formatiert. Ersetzen Sie unter Windows den umgekehrten Schrägstrich (`\`), das Unix-Fortsetzungszeichen, am Ende jeder Zeile durch ein Caret-Zeichen oder Zirkumflex (`^`).

```
aws s3control create-banner-access-point \  
  --region region \  
  --account-id account-id \  
  --name s3-object-lambda-access-point \  
  --configuration file://config.json
```

## Aufrufen eines Amazon S3-Objekt-Lambda-Zugriffspunkts zur Steuerung des Zugriffs auf Dokumente

Im folgenden Beispiel wird ein Amazon S3 Object Lambda Access Point aufgerufen, um den Zugriff auf Dokumente zu steuern.

Aufrufen eines Amazon S3-Objekt-Lambda-Zugriffspunkts mit der AWS Command Line Interface

Im folgenden Beispiel wird ein Amazon S3 Object Lambda Access Point mit der aufgerufen AWS CLI.

Das Beispiel ist für Unix, Linux und macOS formatiert. Ersetzen Sie unter Windows den umgekehrten Schrägstrich (\), das Unix-Fortsetzungszeichen, am Ende jeder Zeile durch ein Caret-Zeichen oder Zirkumflex (^).

```
aws s3api get-object \  
  --region region \  
  --bucket s3-object-lambda-access-point-name-arn \  
  --key object-prefix-key output-file-name
```

## Redigieren von persönlich identifizierbaren Informationen (PII) aus Dokumenten

Sie können einen Amazon S3 Object Lambda Access Point verwenden, um persönlich identifizierbare Informationen (PII) aus Dokumenten zu redigieren.

Um PII-Entitätstypen aus Dokumenten zu redigieren, die in einem S3-Bucket gespeichert sind, verwenden Sie die `ComprehendPiiRedactionS3ObjectLambda` Funktion. Diese Lambda-Funktion verwendet die [DetectPiiEntities](#) Operationen [ContainsPiiEntities](#) und , wenn eine standardmäßige Amazon S3-GET-Anforderung für Dokumentobjekte verarbeitet wird.

Wenn Dokumente in Ihrem S3-Bucket beispielsweise PII wie Kreditkartennummern oder Bankkontoinformationen enthalten, können Sie die `ComprehendPiiRedactionS3ObjectLambda` Funktion so konfigurieren, dass PII erkannt werden, und dann eine Kopie dieser Dokumente

zurückgeben, in denen PII-Entitätstypen redigiert werden. Weitere Informationen zu unterstützten PII-Entitätstypen finden Sie unter [Universelle PII-Entitätstypen](#).

Um weitere Informationen zu dieser Lambda-Funktion zu erhalten, melden Sie sich bei der an, AWS Management Console um die [ComprehendPiiRedactionS3ObjectLambda](#)-Funktion in der anzuzeigenAWS Serverless Application Repository.

## Erstellen eines Lambda-Zugriffspunkts für Amazon S3-Objekte zum Redigieren von PII aus Dokumenten

Im folgenden Beispiel wird ein Amazon S3 Object Lambda Access Point erstellt, um Kreditkartennummern aus Dokumenten zu redigieren.

Erstellen eines Lambda-Zugriffspunkts für Amazon S3-Objekte mit der AWS Command Line Interface

Erstellen Sie eine Konfiguration des Amazon S3 Object Lambda Access Point und speichern Sie die Konfiguration in einer Datei namens `config.json`.

```
{
  "SupportingAccessPoint": "s3-default-access-point-name-arn",
  "TransformationConfigurations": [
    {
      "Actions": [
        "s3:GetObject"
      ],
      "ContentTransformation": {
        "AwsLambda": {
          "FunctionArn": "comprehend-pii-redaction-s3-object-lambda-arn",
          "FunctionPayload": "{\"pii_entities_types\": \"CREDIT_DEBIT_NUMBER\"}"
        }
      }
    }
  ]
}
```

Das folgende Beispiel zeigt das Erstellen eines Amazon S3 Object Lambda Access Point basierend auf der Konfiguration, die in der definiert ist. `config.json`

Das Beispiel ist für Unix, Linux und macOS formatiert. Ersetzen Sie unter Windows den umgekehrten Schrägstrich (\), das Unix-Fortsetzungszeichen, am Ende jeder Zeile durch ein Caret-Zeichen oder Zirkumflex (^).

```
aws s3control create-access-point-for-object-lambda \  
  --region region \  
  --account-id account-id \  
  --name s3-object-lambda-access-point \  
  --configuration file://config.json
```

## Aufrufen eines Lambda-Zugriffspunkts für Amazon S3-Objekte, um PII aus Dokumenten zu redigieren

Die folgenden Beispiele rufen einen Amazon S3 Object Lambda Access Point auf, um PII aus Dokumenten zu redigieren.

Aufrufen eines Amazon S3-Objekt-Lambda-Zugriffspunkts mit der AWS Command Line Interface

Im folgenden Beispiel wird ein Amazon S3 Object Lambda Access Point mit der aufgerufen AWS CLI.

Das Beispiel ist für Unix, Linux und macOS formatiert. Ersetzen Sie unter Windows den umgekehrten Schrägstrich (\), das Unix-Fortsetzungszeichen, am Ende jeder Zeile durch ein Caret-Zeichen oder Zirkumflex (^).

```
aws s3api get-object \  
  --region region \  
  --bucket s3-object-lambda-access-point-name-arn \  
  --key object-prefix-key output-file-name
```

## Lösung: Analysieren von Text mit Amazon Comprehend und OpenSearch

AWS bietet eine Referenzimplementierung der Textanalyse mit Amazon Comprehend und dem OpenSearch Service. Amazon Comprehend bietet Textanalyse und OpenSearch Indizierung, Suche und Visualisierung von Dokumenten.

Weitere Informationen finden Sie unter [Analysieren von Text mit OpenSearch und Amazon Comprehend](#).



# API-Referenz

Die API-Referenz ist jetzt ein separates Dokument. Weitere Informationen finden Sie in [der Amazon Comprehend API-Referenz](#).

# Dokumentverlauf für Amazon Comprehend

In der folgenden Tabelle wird die Dokumentation für diese Version von Amazon Comprehend beschrieben.

Änderung	Beschreibung	Datum
<a href="#">Benutzerdefiniertes Classifier-Training mit nativen Dokumenten</a>	Amazon Comprehend unterstützt jetzt benutzerdefiniertes Classifier-Training mit nativen Dokumenten. Weitere Informationen finden Sie unter <a href="#">Modelle zur Trainingsklassifizierung in Amazon Comprehend</a> .	19. April 2023
<a href="#">Flywheels für die Verwaltung benutzerdefinierter Modelle</a>	Amazon Comprehend unterstützt jetzt Flywheels, um Sie bei der Verwaltung des Trainings und der Nachverfolgung von Modellversionen für benutzerdefinierte Modelle zu unterstützen. Weitere Informationen finden Sie unter <a href="#">Flywheels in Amazon Comprehend</a> .	28. Februar 2023
<a href="#">Aktualisierte IAM-Sicherheitsthemen</a>	Die IAM-Sicherheitsthemen wurden aktualisiert, um Verbundidentitäten einzubeziehen. Weitere Informationen finden Sie unter <a href="#">Identity and Access Management für Amazon Comprehend</a> .	22. Dezember 2022

[Einstufige Verarbeitung für Inferenzen mit benutzerdefinierten Modellen](#)

Amazon Comprehend führt jetzt automatisch die Textextraktion für Bild-, PDF- oder Word-Eingabedokumente durch, bevor eine benutzerdefinierte Klassifizierung oder eine benutzerdefinierte Entitätserkennung ausgeführt wird. Weitere Informationen finden Sie unter [Dokumente verarbeiten in Amazon Comprehend](#).

01. Dezember 2022

[Synchrone APIs für gezielte Stimmung](#)

Amazon Comprehend unterstützt jetzt synchrone APIs und Echtzeitanalysen der Konsole für eine gezielte Stimmung. Die gezielte Stimmung bestimmt die Stimmung, die bestimmten Entitäten in einem Dokument zugeordnet ist. Weitere Informationen finden Sie unter [Gezielte Stimmung in Amazon Comprehend](#).

21. September 2022

### [Niedrigere Mindestanforderungen für Trainingserkennungen](#)

Amazon Comprehend hat die Mindestanforderungen für das Training eines Recognizers mit Klartext-CSV-Anmerkungsdokumenten reduziert. Sie können jetzt ein benutzerdefiniertes Entitätserkennungsmodell mit nur drei kommentierten Dokumenten und mindestens 25 Anmerkungen pro Entitätstyp erstellen. Weitere Informationen finden Sie unter [Vorbereiten der Trainingsdaten](#).

03. August 2022

### [Größere Eingabedokumentgröße für Echtzeit-APIs](#)

Amazon Comprehend unterstützt jetzt bis zu 100KB Eingabedokument für die meisten Echtzeit-APIs. Weitere Informationen finden Sie unter [Richtlinien und Kontingente](#).

18. Juli 2022

### [Zusätzliche PII-Entitätstypen](#)

Zusätzliche PII-Entitätstypen, die jetzt von Amazon Comprehend erkannt werden. Weitere Informationen finden Sie unter [Erkennen von PII-Entitäten in Amazon Comprehend](#).

20. Mai 2022

## [Inhaltsverzeichnis-Neustrukturierung](#)

Das Amazon Comprehend-Entwicklerhandbuch wurde zur einfacheren Navigation umstrukturiert. Weitere Informationen finden Sie unter [Was ist Amazon Comprehend?](#).

7. April 2022

## [Gezielte Stimmung](#)

Amazon Comprehend unterstützt jetzt eine gezielte Stimmungsanalyse, die die Stimmung bestimmt, die bestimmten Entitäten in einem Dokument zugeordnet ist. Weitere Informationen finden Sie unter [Gezielte Stimmung in Amazon Comprehend](#).

9. März 2022

## [Neues Feature](#)

Mit Amazon Comprehend können Sie jetzt Bilder auf benutzerdefinierte Berechtigungserkennung analysieren. Weitere Informationen finden Sie unter [Erkennen von benutzerdefinierten Entitäten in Amazon Comprehend](#).

28. Februar 2022

## [Neues Feature](#)

Sie können jetzt trainierte benutzerdefinierte Modelle zwischen kopierenAWS-Konten. Weitere Informationen finden Sie unter [Kopieren von benutzerdefinierten Modellen zwischen Konten in Amazon Comprehend](#).

2. Februar 2022

[Neues Feature](#)

Sie können jetzt verwenden , AWS Trusted Advisor um Empfehlungen anzuzeigen, mit denen Sie die Kosten und Sicherheit Ihrer Amazon Comprehend-Endpunkte optimieren können. Weitere Informationen finden Sie unter [Verwenden von Trusted Advisor mit Amazon Comprehend.](#)

29. September 2021

[Neues Feature](#)

Amazon Comprehend hat eine Reihe von Funktionen für Comprehend Custom eingeführt, die kontinuierliche Modellverbesserungen ermöglichen, indem Sie neue Modellversionen erstellen, kontinuierlich an bestimmten Testsätzen testen und eine Live-Migration zu neuen Modellendpunkten durchführen können.

21. September 2021

[Neues Feature](#)

Mit Amazon Comprehend können Sie jetzt PDF- und Word-Dokumente auf benutzerdefinierte Berechtigungserkennung analysieren. Mit PDF- und Word-Formaten können Sie Informationen aus Dokumenten extrahieren, die Header, Listen und Tabellen enthalten.

14. September 2021

## Neues Feature

Amazon Comprehend hat eine neue Endpunktübersichtsfunktion eingeführt, die Ihnen einen globalen Überblick über Ihre Endpunkte bietet. Auf der Übersichtsseite Endpunkte können Sie alle Endpunkte an einem Ort anzeigen, um Ihre Endpunktnutzung im Vergleich zu Ihrer tatsächlichen Ressourcennutzung zu verstehen.

24. August 2021

## Neues Feature

Mit Amazon Comprehend Medical können Sie jetzt eine private Verbindung zu Ihrer Virtual Private Cloud (VPC) herstellen, indem Sie einen Schnittstellen-VPC-Endpunkt erstellen. Weitere Informationen finden Sie unter [VPC-Endpunkte \(PrivateLink\)](#).

13. Juni 2021

## Spracherweiterung

Amazon Comprehend hat vier zusätzliche Sprachen für das dominante Sprachfeature hinzugefügt: Hausa (ha), Lao (lo), Maltesisch (mt) und Oromo (om). Weitere Informationen finden Sie unter [Unterstützte Sprachen in Amazon Comprehend](#).

10. Mai 2021

## Neues Feature

Mit Amazon Comprehend können Sie jetzt benutzerdefinierte Modelle mit einem vom Kunden verwalteten Schlüssel (CMK) verschlüsseln. Weitere Informationen finden Sie unter [KMS-Verschlüsselung in Amazon Comprehend](#).

31. März 2021

## Neues Feature

Sie können jetzt Amazon S3 Object Lambda Access Points verwenden, um zu konfigurieren, wie Dokumente, die persönlich identifizierbare Informationen (PII) enthalten, aus Ihrem Amazon S3-Bucket abgerufen werden. Sie können den Zugriff auf Dokumente steuern, die PII enthalten, und PII aus Dokumenten redigieren. Weitere Informationen finden Sie unter [Verwenden von Amazon S3-Objekt-Lambda-Zugriffspunkten für persönlich identifizierbare Informationen \(PII\)](#).

18. März 2021



## Neues Feature

Sie können jetzt ein Dokument mit persönlich identifizierbaren Informationen (PII) kennzeichnen. Amazon Comprehend kann Ihr Dokument auf das Vorhandensein von PII analysieren und die Labels identifizierter PII-Entitätstypen wie Name, Adresse, Bankkontonummer oder Telefonnummer zurückgeben. Weitere Informationen finden Sie unter [Dokument mit PII kennzeichnen](#).

11. März 2021

## Neues Feature

Mit Amazon Comprehend können Sie jetzt Ereignisse in einer Reihe von Dokumenten erkennen. Wenn Sie einen asynchronen Ereigniserkennungsauftrag erstellen, kann Amazon Comprehend unterstützte Arten von Finanzereignissen erkennen. Weitere Informationen finden Sie unter [Erkennen von Ereignissen](#).

24. November 2020

## Neues Feature

Mit Amazon Comprehend können Sie jetzt Auto Scaling für benutzerdefinierte Entity Recognizer-Endpunkte verwenden. Mit Auto Scaling können Sie die Endpunktbereitstellung automatisch an Ihre Kapazitätsanforderungen anpassen. Weitere Informationen finden Sie unter [Auto Scaling mit Endpunkten](#).

28. September 2020

## Neues Feature

Um benutzerdefinierte Classifier oder Entity Recognizer zu trainieren, können Sie jetzt erweiterte Manifestdateien bereitstellen, bei denen es sich um beschriftete Datensätze handelt, die von Amazon SageMaker Ground Truth erstellt werden. Weitere Informationen zu diesen Dateien und Beispiele finden Sie unter [Mehrklassenmodus, Multi-Label-Modus](#) und [Anmerkungen](#).

22. September 2020

## [Neues Tutorial](#)

Amazon Comprehend bietet jetzt ein Tutorial, das Sie durch einen Multi-Service-Workflow führt, bei dem Sie Kundenüberprüfungen analysieren und die Analyseergebnisse visualisieren. Weitere Informationen finden Sie unter [Tutorial: Analysieren von Erkenntnissen aus Bewertungen](#).

17. September 2020

## [Neues Feature](#)

Mit Amazon Comprehend können Sie jetzt Entitäten in Ihrem Text erkennen, die persönlich identifizierbare Informationen (PII) wie Adressen, Bankkontonummern oder Telefonnummern enthalten. Amazon Comprehend kann den Speicherort jeder PII-Entität in Ihrem Text oder eine Kopie Ihres Textes bereitstellen, in dem die PII redigiert werden. Weitere Informationen finden Sie unter [Erkennen von persönlich identifizierbaren Informationen \(PII\)](#).

17. September 2020

## Neues Feature

Bisher konnten Sie ein Modell nur auf bis zu 12 benutzerdefinierten Entitäten trainieren. Jetzt können Sie mit Amazon Comprehend ein Modell auf bis zu 25 benutzerdefinierten Entitäten gleichzeitig trainieren. Weitere Informationen finden Sie unter [Benutzerdefinierte Entitätserkennung](#).

12. August 2020

## Spracherweiterung

Amazon Comprehend hat fünf zusätzliche Sprachen für die Funktion zur Erkennung benutzerdefinierter Entitäten hinzugefügt: Deutsch (de), Spanisch (es), Französisch (fr), Italienisch (it) und Portugiesisch (pt). Weitere Informationen finden Sie unter [Unterstützte Sprachen in Amazon Comprehend](#).

12. August 2020

## Neues Feature

Mit Amazon Comprehend können Sie jetzt eine private Verbindung zu Ihrer Virtual Private Cloud (VPC) herstellen, indem Sie einen Schnittstellen-VPC-Endpunkt erstellen. Weitere Informationen finden Sie unter [VPC-Endpunkte \(AWS PrivateLink\)](#).

11. August 2020

## Neues Feature

Mit Amazon Comprehend können Sie jetzt benutzerdefinierte Entitäten in einzelnen Textdokumenten schnell erkennen, indem Sie Echtzeitanalysen durchführen. Weitere Informationen finden Sie unter [Erkennen von benutzerdefinierten Entitäten in Echtzeit mit Amazon Comprehend](#).

9. Juli 2020

## Neue Funktion hinzugefügt

Amazon Comprehend bietet jetzt Unterstützung für einen zweiten Modus in asynchroner benutzerdefinierter Klassifizierung für Dokumente, die mehr Flexibilität bei der Anwendung benutzerdefinierter Klassen auf Dokumente bieten. Während der Mehrklassenmodus jedem Dokument nur eine einzelne Klasse zuordnet, kann der neue Multi-Label-Modus mehr als eine zuordnen. Beispielsweise kann ein Film gleichzeitig sowohl als wissenschaftliche Fiction als auch als Aktion klassifiziert werden. Weitere Informationen finden Sie unter [Modi mit mehreren Klassen und Bezeichnungen in der benutzerdefinierten Klassifizierung](#).

19. Dezember 2019

### Neue Funktion hinzugefügt

Amazon Comprehend bietet jetzt Unterstützung für benutzerdefinierte Echtzeitklassifizierung für Dokumente mit unstrukturiertem Text. Kunden können die benutzerdefinierte Echtzeitklassifizierung verwenden, um Informationen basierend auf ihren eigenen Geschäftsregeln synchron zu verstehen, zu kennzeichnen und weiterzuleiten. Weitere Informationen finden Sie unter [Echtzeitanalyse mit benutzerdefinierter Klassifizierung](#).

25. November 2019

### Neue Sprachen hinzugefügt

Amazon Comprehend hat sechs zusätzliche Sprachen für mehrere seiner Funktionen hinzugefügt: Arabisch (Ar), Hindi (Hi), Japanisch (ja), Koreanisch (ko), vereinfachtes Chinesisch (zh) und traditionelles Chinesisch (zh-). Diese neuen Sprachen werden nur für die Operationen „Stimmung bestimmen“, „Schlüsselphrasen erkennen“ und „Nicht benutzerdefinierte Detect-Entitäten“ unterstützt. Weitere Informationen finden Sie unter [Unterstützte Sprachen](#).

6. November 2019

## Neues Feature

Bisher konnten Sie ein Modell nur auf einer einzigen benutzerdefinierten Entität trainieren. Daher könnten Sie nur nach dieser Entität mit einer Entitätserkennung operation suchen. Amazon Comprehend hat dies geändert und Sie können jetzt ein Modell mit bis zu 12 benutzerdefinierten Entitäten gleichzeitig trainieren. Weitere Informationen finden Sie unter [Benutzerdefinierte Entitätserkennung](#).

9. Juli 2019

## Neues Feature

Amazon Comprehend bietet jetzt eine Konfusionsmatrix mit mehreren Klassen für die zusätzliche Möglichkeit, Metriken beim Training eines benutzerdefinierten Classifiers zu analysieren. Dies wird derzeit nur unter Verwendung der -APIs unterstützt. Weitere Informationen finden Sie unter [Markieren von Ressourcen in Amazon Comprehend](#)

5. April 2019

## Neues Feature

Amazon Comprehend bietet Tags für benutzerdefinierte Classifier und benutzerdefinierte Entitätserkennung, die als Metadaten verwendet werden können, mit denen Sie den Zugriff auf Ihre Ressourcen mit einem feineren Maß an Kontrolle als je zuvor organisieren, filtern und steuern können. Weitere Informationen finden Sie unter [Markieren von Ressourcen in Amazon Comprehend](#)

3. April 2019

## Neues Feature

Mit Amazon S3 können Sie Ihre Eingabedokumente bereits verschlüsseln, und Amazon Comprehend erweitert dies noch weiter. Durch die Verwendung Ihres eigenen KMS-Schlüssels können Sie nicht nur die Ausgabeergebnisse Ihres Auftrags verschlüsseln, sondern auch die Daten auf dem Speicher-Volumen, das der Rechen-Instance zugeordnet ist, die den Analyseauftrag verarbeitet. Das Ergebnis ist end-to-end Sicherheit. Weitere Informationen finden Sie unter [KMS-Verschlüsselung in Amazon Comprehend](#).

28. März 2019



### Neues Feature

Die Erkennung benutzerdefinierter Entitäten erweitert die Funktion von Amazon Comprehend, indem Sie neue Entitätstypen identifizieren können, die nicht als einer der voreingestellten generischen Entitätstypen unterstützt werden. Das bedeutet, dass Sie Dokumente analysieren und Entitäten wie Produktcodes oder geschäftsspezifische Entitäten extrahieren können, die Ihren speziellen Anforderungen entsprechen. Weitere Informationen finden Sie unter [Benutzerdefinierte Entitätserkennung](#).

16. November 2018

### Neues Feature

Sie können Amazon Comprehend verwenden, um Ihre eigenen Modelle für die benutzerdefinierte Klassifizierung zu erstellen und ein Dokument einer Klasse oder Kategorie zuzuweisen. Weitere Informationen finden Sie unter [Dokumentenklassifizierung](#).

15. November 2018

### Regionale Erweiterung

Amazon Comprehend ist jetzt in Europa (Frankfurt) (eu-central-1) verfügbar.

10. Oktober 2018

---

<a href="#">Spracherweiterung</a>	Zusätzlich zu Englisch und Spanisch kann Amazon Comprehend jetzt auch Dokumente auf Französisch, Deutsch, Italienisch und Portugiesisch untersuchen. Weitere Informationen finden Sie unter <a href="#">Unterstützte Sprachen in Amazon Comprehend</a> .	10. Oktober 2018
<a href="#">Regionale Erweiterung</a>	Amazon Comprehend ist jetzt in Asien-Pazifik (Sydney) (ap-southeast-2) verfügbar.	15. August 2018
<a href="#">Neues Feature</a>	Amazon Comprehend analysiert jetzt Dokumente, um die Syntax eines Dokuments und den Sprachteil für jedes Wort zu ermitteln. Weitere Informationen finden Sie unter <a href="#">Syntax</a> .	17. Juli 2018
<a href="#">Neues Feature</a>	Amazon Comprehend unterstützt jetzt die asynchrone Batchverarbeitung für die Sprach-, Schlüsselphrase-, Entitäts- und Stimmungserkennung. Weitere Informationen finden Sie unter <a href="#">Asynchrone Batchverarbeitung</a> .	27. Juni 2018
<a href="#">Neues Handbuch</a>	Dies ist die erste Version des Amazon Comprehend-Entwicklerhandbuchs.	29. November 2017

# AWS-Glossar

Die neueste AWS-Terminologie finden Sie im [AWS-Glossar](#) in der AWS-Glossar-Referenz.

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.