Referenzhandbuch

AWS SDKs und Tools

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

AWS SDKs und Tools: Referenzhandbuch

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

Die Handelsmarken und die 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

AWS Referenzhandbuch für SDKs und Tools 1
Ressourcen für Entwickler 3
Telemetrie-Benachrichtigung im Toolkit 3
Konfiguration 5
Geteilte `credentials` Dateien `config` und Dateien 6
Profile 6
Format der Konfigurationsdatei 8
Format der Datei mit den Anmeldeinformationen 11
Speicherort der geteilten Dateien 12
Auflösung des Home-Verzeichnisses 12
Ändern Sie den Standardspeicherort dieser Dateien 13
Umgebungsvariablen 14
Festlegen von Umgebungsvariablen 15
Einrichtung von serverlosen Umgebungsvariablen 16
JVM-Systemeigenschaften 17
Wie legt man die JVM-Systemeigenschaften fest 17
Authentifizierung und Zugriff 20
Wählen Sie eine Methode zur Authentifizierung Ihres Anwendungscodes 20
Authentifizierungsmethoden 24
AWS Builder ID 26
Melden Sie sich mit Anmeldeinformationen für die Konsole an 27
Funktionsweise 27
Authentifizierung von IAM Identity Center 28
Voraussetzungen 28
Konfigurieren Sie den programmatischen Zugriff mit IAM Identity Center 29
Aktualisierung der Portalzugriffssitzungen 32
Verstehen Sie die IAM Identity Center-Authentifizierung 32
IAM Roles Anywhere 37
Schritt 1: IAM-Rollen Anywhere konfigurieren 37
Schritt 2: Verwenden Sie IAM Roles Anywhere 37
Übernehmen einer Rolle 39
Nehmen Sie eine IAM-Rolle an 39
Nehmen Sie eine Rolle an (Web) 41
Verbunden mit Web-Identität oder OpenID Connect 42

AWS Zugriffstasten ... 44
Verwenden Sie kurzfristige Anmeldeinformationen ... 44
Verwenden Sie langfristige Anmeldeinformationen ... 44
Kurzfristige Anmeldeinformationen ... 46
Langfristige Anmeldeinformationen ... 47
IAM-Rollen für Instances EC2 ... 51
Erstellen einer IAM-Rolle ... 51
Starten Sie eine EC2 Amazon-Instance und geben Sie Ihre IAM-Rolle an ... 52
Connect zur EC2 Instanz her ... 52
Führen Sie Ihre Anwendung auf der Instanz aus EC2 ... 53
Weitergabe von vertrauenswürdigen Identitäten ... 53
Voraussetzungen für die Verwendung des TIP-Plug-ins ... 54
Um das TIP-Plugin in Ihrem Code zu verwenden ... 55
Codebeispiele mit TIP ... 58
Referenz zu Einstellungen ... 65
Serviceclients erstellen ... 65
Vorrang der Einstellungen ... 65
Die Einstellungsseiten dieses Handbuchs verstehen ... 67
`Config`Liste der Dateieinstellungen ... 68
`Credentials`Liste der Dateieinstellungen ... 73
Liste der Umgebungsvariablen ... 74
Liste der JVM-Systemeigenschaften ... 78
Standardisierte Anbieter von Anmeldeinformationen ... 82
Machen Sie sich mit der Kette der Anbieter von Anmeldeinformationen vertraut ... 83
SDK-specific und toolspezifische Anbieterketten für Anmeldeinformationen ... 85
AWS Zugriffstasten ... 85
Anbieter für Anmeldedaten ... 89
Nehmen Sie die Rolle des Anbieters an ... 91
Container-Anbieter ... 99
IAM Identity Center-Anbieter ... 103
IMDS-Anbieter ... 111
Prozessanbieter ... 116
Standardisierte Funktionen ... 121
Account-based-Endpunkte ... 122
Anwendungs-ID ... 125
Amazon EC2-Instance-Metadaten ... 127

Amazon S3 Access Points ... 130
Amazon S3 Multi-Region S3-Zugriffspunkte ... 133
S3 Express One Zone-Sitzungsauthentifizierung ... 135
Authentifizierungsschema ... 138
AWS-Region ... 141
AWS STS Regionale Endpunkte ... 144
Schutz der Datenintegrität ... 150
Dual-stack und FIPS-Endpunkte ... 155
Endpunkterkennung ... 158
Allgemeine Konfiguration ... 160
Host-Präfix-Injektion ... 164
IMDS-Kunde ... 169
Wiederholungsverhalten ... 172
Komprimierung anfordern ... 188
Service-specific Endpunkte ... 190
Standardeinstellungen für intelligente Konfigurationen ... 237
Paginierung ... 243
Sicherheit ... 246
Aktivierung von hybridem Post-Quantum-TLS ... 246
SDKs mit standardmäßig aktiviertem PQ TLS ... 247
Opt-in PQ TLS-Unterstützung ... 248
SDKs, die auf System OpenSSL basieren ... 249
AWS SDKs und Tools planen nicht, PQ TLS zu unterstützen ... 251
Allgemeine Runtime ... 252
CRT-Abhängigkeiten ... 253
Wartungsrichtlinie ... 254
-Übersicht ... 254
Versionsverwaltung ... 254
Lebenszyklus der SDK-Hauptversionen ... 254
Lebenszyklus von Abhängigkeiten ... 255
Methoden der Kommunikation ... 256
Lebenszyklus der Version ... 258
Dokumentverlauf ... 261
... cclxv

Was ist abgedeckt in der AWS Referenzhandbuch für SDKs und Tools

Viele SDKs und Tools weisen einige gemeinsame Funktionen auf, entweder durch gemeinsame Designspezifikationen oder durch eine gemeinsame Bibliothek.

Dieses Handbuch enthält Informationen zu:

- Weltweite Konfiguration AWS SDKs und Tools— Wie Sie die Variablen shared `config` und `credentials` files oder environment verwenden, um Ihre AWS SDKs und Tools zu konfigurieren.
- Authentifizierung und Zugriff mithilfe von AWS SDKs Tools— Stellen Sie fest, wie sich Ihr Code oder Tool authentifiziert AWS , wenn Sie mit entwickeln. AWS-Services
- AWS SDKs und Referenz zu den Werkzeugeinstellungen— Referenz für alle standardisierten Einstellungen, die für die Authentifizierung und Konfiguration verfügbar sind.
- AWS Common Runtime (CRT) -Bibliotheken— Überblick über die gemeinsam genutzten AWS Common Runtime (CRT) -Bibliotheken, die für fast alle SDKs verfügbar sind.
- AWS Wartungsrichtlinie für SDKs und Toolsbehandelt die Wartungsrichtlinien und die Versionierung für AWS Software Development Kits (SDKs) und Tools, einschließlich SDKs für mobile Geräte und Internet der Dinge (IoT), sowie die zugrunde liegenden Abhängigkeiten.

Dieses Referenzhandbuch für AWS SDKs und Tools soll als Informationsbasis dienen, die für mehrere SDKs und Tools gilt. Das spezifische Handbuch für das von Ihnen verwendete SDK oder Tool sollte zusätzlich zu den hier aufgeführten Informationen verwendet werden. Im Folgenden finden Sie das SDK und die Tools, die relevante Abschnitte des Materials in diesem Handbuch enthalten:

Wenn Sie Folgendes verwenden:	Die für Sie relevanten Abschnitte dieses Handbuchs sind:
• Jedes SDK oder Tool	AWS Wartungsrichtlinie für SDKs und Tools
• AWS Cloud9 • AWS CDK • AWS Toolkit für Azure DevOps • AWS Toolkit for JetBrains	Weltweite Konfiguration AWS SDKs und Tools Authentifizierung und Zugriff mithilfe von AWS SDKs Tools AWS Wartungsrichtlinie für SDKs und Tools

Wenn Sie Folgendes verwenden:	Die für Sie relevanten Abschnitte dieses Handbuchs sind:
• AWS Toolkit for Visual Studio • AWS Toolkit for Visual Studio Code • AWS Serverless Application Model • AWS CodeArtifact • AWS CodeBuild • Amazon CodeCatalyst • AWS CodeCommit • AWS CodeDeploy • AWS CodePipeline	
• AWS CLI • AWS SDK für C++ • AWS SDK für Go • AWS SDK für Java • AWS SDK für JavaScript • AWS SDK für Kotlin • AWS SDK für .NET • AWS SDK für PHP • AWS SDK für Python (Boto3) • AWS SDK für Ruby • AWS SDK für Rust • AWS SDK für Swift • AWS Tools for Windows PowerShell	Weltweite Konfiguration AWS SDKs und Tools Authentifizierung und Zugriff mithilfe von AWS SDKs Tools AWS SDKs und Referenz zu den Werkzeuge instellungen AWS Common Runtime (CRT) -Bibliotheken AWS Wartungsrichtlinie für SDKs und Tools AWS Versionslebenszyklus von SDKs und Tools

- Einen Überblick über Tools, die Ihnen bei der Entwicklung von Anwendungen helfen können, finden Sie unter Tools AWS, auf denen Sie aufbauen können AWS.
- Informationen zum Support finden Sie im AWS Knowledge Center.
- Die AWS Terminologie finden Sie im AWS Glossar in der AWS-Glossar Referenz.

Ressourcen für Entwickler

Amazon Q Developer ist ein generativer AI-powered Konversationsassistent, der Ihnen helfen kann, AWS Anwendungen zu verstehen, zu erstellen, zu erweitern und zu bedienen. Damit Sie schneller darauf aufbauen können AWS, wird das Modell, das Amazon Q zugrunde liegt, um qualitativ hochwertige AWS Inhalte erweitert, um vollständigere, umsetzbarere und referenziertere Antworten zu erhalten. Weitere Informationen finden Sie unter Was ist Amazon Q Developer? im Benutzerhandbuch zu Amazon Q Developer.

Telemetrie-Benachrichtigung im Toolkit

AWS IDE-Toolkits (Integrated Development Environment) sind Plugins und Erweiterungen, die den Zugriff auf AWS Dienste in Ihrer IDE ermöglichen. Amazon Q IDE-Plug-ins und -Erweiterungen ermöglichen generative KI-Unterstützung in Ihrer IDE. Detaillierte Informationen zu den einzelnen IDE-Toolkits finden Sie in den Toolkit-Benutzerhandbüchern in der vorherigen Tabelle. Weitere Informationen zur Verwendung von Amazon Q in Ihrer IDE finden Sie im Thema Verwenden von Amazon Q in der IDE im Amazon Q-Entwicklerhandbuch.

AWS IDE Toolkits und Amazon Q können clientseitige Telemetriedaten sammeln und speichern, um Entscheidungen über future AWS Toolkit- und Amazon Q-Versionen zu treffen. Die gesammelten Daten quantifizieren Ihre Nutzung des AWS Toolkits und von Amazon Q.

Weitere Informationen zu den Telemetriedaten, die in allen AWS IDE-Toolkits und Amazon Q gesammelt wurden, finden Sie im gemeinsamen Definitions.json Dokument im `aws-toolkit-common` Github-Repository.

Detaillierte Informationen zu den Telemetriedaten, die von den einzelnen AWS IDE-Toolkits und Amazon Q-Erweiterungen gesammelt werden, finden Sie in den Ressourcendokumenten in den folgenden AWS GitHub Toolkit-Repositorys:

- AWS Visual Studio Toolkit mit Amazon Q
- AWS Toolkit for Visual Studio Code und Amazon Q-Erweiterung für VS Code
- AWS Toolkit for JetBrains und Amazon Q-Plugin für JetBrains
- Amazon Q für Eclipse

Bestimmte AWS Dienste, auf die in den AWS Toolkits zugegriffen werden kann, können zusätzliche clientseitige Telemetriedaten sammeln. Detaillierte Informationen über die Art der Daten, die von

den einzelnen AWS Diensten erfasst werden, finden Sie im Thema AWS Dokumentation für den jeweiligen Dienst, an dem Sie interessiert sind.

Weltweite Konfiguration AWS SDKs und Tools

Mit AWS SDKs und anderen AWS Entwicklertools wie dem AWS Command Line Interface (AWS CLI) können Sie mit dem AWS Service interagieren APIs. Bevor Sie dies versuchen, müssen Sie das SDK oder das Tool jedoch mit den Informationen konfigurieren, die es für die Ausführung des angeforderten Vorgangs benötigt.

Diese Informationen umfassen die folgenden Elemente:

- Informationen zu Anmeldeinformationen, anhand derer identifiziert wird, wer die API aufruft. Die Anmeldeinformationen werden verwendet, um die Anfrage an die AWS Server zu verschlüsseln. Anhand dieser Informationen wird Ihre Identität AWS bestätigt und die zugehörigen Berechtigungsrichtlinien können abgerufen werden. Dann kann es bestimmen, welche Aktionen Sie ausführen dürfen.
- Andere Konfigurationsdetails, anhand derer Sie dem AWS CLI SDK mitteilen, wie die Anfrage verarbeitet werden soll, wohin die Anfrage gesendet werden soll (an welchen AWS Dienstendpunkt) und wie die Antwort interpretiert oder angezeigt werden soll.

Jedes SDK oder Tool unterstützt mehrere Quellen, über die Sie die erforderlichen Anmeldeinformationen und Konfigurationsinformationen bereitstellen können. Einige Quellen sind nur für das SDK oder Tool verfügbar. Einzelheiten zur Verwendung dieser Methode finden Sie in der Dokumentation zu diesem Tool oder SDK.

Die Tools AWS SDKs und und unterstützen jedoch allgemeine Einstellungen aus Primärquellen, die über den Code selbst hinausgehen. Dieser Abschnitt deckt die folgenden Themen ab:

Themen

- Verwenden von geteilten credentials Dateien config und Dateien zur globalen Konfiguration AWS SDKs und Tools
- Suchen und Ändern des Speicherorts der geteilten credentials Dateien config AWS SDKs und Tools
- Verwendung von Umgebungsvariablen zur globalen Konfiguration AWS SDKs und Tools
- Verwenden von JVM-Systemeigenschaften zur globalen Konfiguration AWS SDK für Java und AWS SDK für Kotlin

Verwenden von geteilten `credentials` Dateien `config` und Dateien zur globalen Konfiguration AWS SDKs und Tools

Die gemeinsam genutzten `credentials` Dateien AWS `config` und Dateien sind die gängigste Methode, um die Authentifizierung und Konfiguration für ein AWS SDK oder Tool festzulegen.

Die gemeinsam genutzten `credentials` Dateien `config` und Dateien enthalten eine Reihe von Profilen. Ein Profil ist ein Satz von Konfigurationseinstellungen in Schlüssel-Wert-Paaren, der von AWS SDKs, the AWS Command Line Interface (AWS CLI) und anderen Tools verwendet wird. Konfigurationswerte werden an ein Profil angehängt, um einen bestimmten Aspekt der SDK/tool Verwendung dieses Profils zu konfigurieren. Diese Dateien werden insofern „gemeinsam genutzt“, als die Werte für alle Anwendungen, Prozesse oder in SDKs der lokalen Umgebung eines Benutzers wirksam werden.

Sowohl die gemeinsam genutzten `config` Dateien als auch die `credentials` Dateien sind Klartextdateien, die nur ASCII-Zeichen (UTF-8-kodiert) enthalten. Sie haben die Form von Dateien, die allgemein als INI-Dateien bezeichnet werden.

Profile

Die Einstellungen in den geteilten `credentials` Dateien `config` und Dateien sind einem bestimmten Profil zugeordnet. In der Datei können mehrere Profile definiert werden, um unterschiedliche Einstellungskonfigurationen für unterschiedliche Entwicklungsumgebungen zu erstellen.

Das `[default]` Profil enthält die Werte, die von einem SDK- oder Tool-Vorgang verwendet werden, wenn kein bestimmtes benanntes Profil angegeben ist. Sie können auch separate Profile erstellen, auf die Sie explizit anhand ihres Namens verweisen können. Jedes Profil kann je nach Anwendung und Szenario unterschiedliche Einstellungen und Werte verwenden.

> **Note**
>
> `[default]`ist einfach ein unbenanntes Profil. Dieses Profil hat seinen Namen`default`, weil es das Standardprofil ist, das vom SDK verwendet wird, wenn der Benutzer kein Profil angibt. Es stellt anderen Profilen keine vererbten Standardwerte zur Verfügung. Wenn Sie im `[default]` Profil etwas festlegen und es nicht in einem benannten Profil festlegen, wird der Wert nicht festgelegt, wenn Sie das benannte Profil verwenden.

Legen Sie ein benanntes Profil fest

Das `[default]` Profil und mehrere benannte Profile können in derselben Datei vorhanden sein. Verwenden Sie die folgende Einstellung, um auszuwählen, welche Profileinstellungen von Ihrem SDK oder Tool bei der Ausführung Ihres Codes verwendet werden. Profile können auch innerhalb des Codes oder pro Befehl ausgewählt werden, wenn Sie mit dem AWS CLI arbeiten.

Konfigurieren Sie diese Funktionalität, indem Sie eine der folgenden Einstellungen festlegen:

`AWS_PROFILE`- Umgebungsvariable

Wenn diese Umgebungsvariable auf ein benanntes Profil oder „Standard“ gesetzt ist, verwenden der gesamte SDK-Code und alle AWS CLI Befehle die Einstellungen in diesem Profil.

Linux/macOS-Beispiel für das Setzen von Umgebungsvariablen über die Befehlszeile:

```
export AWS_PROFILE="my_default_profile_name";
```

Windows-Beispiel für das Setzen von Umgebungsvariablen über die Befehlszeile:

```
setx AWS_PROFILE "my_default_profile_name"
```

`aws.profile`- JVM-Systemeigenschaft

Für das SDK für Kotlin auf der JVM und das SDK for Java 2.x können Sie die `aws.profile` Systemeigenschaft festlegen. Wenn das SDK einen Dienstclient erstellt, verwendet es die Einstellungen im genannten Profil, sofern die Einstellung nicht im Code überschrieben wird. Das SDK for Java 1.x unterstützt diese Systemeigenschaft nicht.

> **Note**
>
> Wenn sich Ihre Anwendung auf einem Server befindet, auf dem mehrere Anwendungen ausgeführt werden, empfehlen wir, immer benannte Profile anstelle des Standardprofils zu verwenden. Das Standardprofil wird automatisch von allen AWS Anwendungen in der Umgebung übernommen und von allen Anwendungen gemeinsam genutzt. Wenn also jemand anderes das Standardprofil für seine Anwendung aktualisiert, kann sich dies unbeabsichtigt auf die anderen auswirken. Um dies zu verhindern, definieren Sie ein benanntes Profil in der gemeinsam genutzten `config` Datei und verwenden Sie dann dieses

benannte Profil in Ihrer Anwendung, indem Sie das benannte Profil in Ihrem Code festlegen. Sie können die Umgebungsvariable oder die JVM-Systemeigenschaft verwenden, um das benannte Profil festzulegen, wenn Sie wissen, dass sich sein Geltungsbereich nur auf Ihre Anwendung auswirkt.

Format der Konfigurationsdatei

Die `config` Datei ist in Abschnitte unterteilt. Ein Abschnitt ist eine benannte Sammlung von Einstellungen und reicht bis zur nächsten Abschnittsdefinitionszeile.

Die `config` Datei ist eine Klartextdatei, die das folgende Format verwendet:

- Alle Einträge in einem Abschnitt haben das allgemeine Format `setting-name=value`.
- Zeilen können auskommentiert werden, indem die Zeile mit einem Hashtag-Zeichen (#) begonnen wird.

Typen von Abschnitten

Eine Abschnittsdefinition ist eine Zeile, die einer Sammlung von Einstellungen einen Namen zuweist. Die Zeilen der Abschnittsdefinition beginnen und enden mit eckigen Klammern (`[]`). Innerhalb der Klammern befinden sich eine Typ-ID für den Abschnitt und ein benutzerdefinierter Name für den Abschnitt. Sie können Buchstaben, Zahlen, Bindestriche () und Unterstriche (`-_`) verwenden, aber keine Leerzeichen.

Abschnittstyp: **`default`**

Beispiel für eine Abschnittsdefinitionszeile: `[default]`

`[default]`ist das einzige Profil, für das die `profile` Abschnitts-ID nicht erforderlich ist.

Das folgende Beispiel zeigt eine `config` Basisdatei mit einem `[default]` Profil. Es legt die `region`Einstellung fest. Alle Einstellungen, die dieser Zeile folgen, sind Teil dieses Profils, bis eine andere Abschnittsdefinition gefunden wird.

```
[default]
#Full line comment, this text is ignored.
region = us-east-2
```

Abschnittstyp: **`profile`**

Beispiel für eine Abschnittsdefinitionszeile: `[profile dev]`

Die `profile` Abschnittsdefinitionszeile ist eine benannte Konfigurationsgruppierung, die Sie für verschiedene Entwicklungsszenarien anwenden können. Weitere Informationen zu benannten Profilen finden Sie im vorherigen Abschnitt über Profile.

Das folgende Beispiel zeigt eine `config` Datei mit einer `profile` Abschnittsdefinitionszeile und einem benannten Profil namens`foo`. Alle Einstellungen, die auf diese Zeile folgen, bis eine andere Abschnittsdefinition gefunden wird, sind Teil dieses benannten Profils.

```
[profile foo]
...settings...
```

Einige Einstellungen haben ihre eigene verschachtelte Gruppe von Untereinstellungen, wie die `s3` Einstellung und die Untereinstellungen im folgenden Beispiel. Ordnen Sie die Untereinstellungen der Gruppe zu, indem Sie sie um ein oder mehrere Leerzeichen einrücken.

```
[profile test]
region = us-west-2
s3 =
    max_concurrent_requests=10
    max_queue_size=1000
```

Abschnittstyp: **`sso-session`**

Beispiel für eine Abschnittsdefinitionszeile: `[sso-session my-sso]`

Die `sso-session` Abschnittsdefinitionszeile benennt eine Gruppe von Einstellungen, die Sie verwenden, um ein Profil für die Auflösung von AWS Anmeldeinformationen zu konfigurieren AWS IAM Identity Center. Weitere Informationen zur Konfiguration der Single Sign-On-Authentifizierung finden Sie unterVerwenden von IAM Identity Center zur Authentifizierung von AWS SDK und Tools. Ein Profil ist mit einem `sso-session` Abschnitt durch ein Schlüssel-Wert-Paar verknüpft, wobei `sso-session` der Schlüssel und der Name Ihres `sso-session` Abschnitts der Wert ist, z. B. `sso-session = <name-of-sso-session-section>`

Im folgenden Beispiel wird ein Profil konfiguriert, das mithilfe eines Tokens von „my-sso“ kurzfristige AWS Anmeldeinformationen für die IAM-Rolle SampleRole "" im Konto „111122223333" erhält. Der

Abschnitt „my-sso“ wird im `sso-session` Abschnitt unter Verwendung des Schlüssels namentlich referenziert. `profile` `sso-session`

```
[profile dev]
sso_session = my-sso
sso_account_id = 111122223333
sso_role_name = SampleRole

[sso-session my-sso]
sso_region = us-east-1
sso_start_url = https://my-sso-portal.awsapps.com/start
```

Abschnittstyp: **`services`**

Beispiel für eine Abschnittsdefinitionszeile: `[services dev]`

> **Note**
>
> Der `services` Abschnitt unterstützt dienstspezifische Endpunktanpassungen und ist nur in SDKs Tools verfügbar, die diese Funktion enthalten. Informationen darüber, ob diese Funktion für Ihr SDK verfügbar ist, finden Sie unter Servicespezifische Support durch AWS SDKs und Tools Endpunkte.

`services`In der Definitionszeile des Abschnitts wird eine Gruppe von Einstellungen benannt, mit denen benutzerdefinierte Endpunkte für Anfragen konfiguriert werden. AWS-Service Ein Profil ist mit einem `services` Abschnitt durch ein Schlüssel-Wert-Paar verknüpft, wobei `services` der Schlüssel und der Name Ihres `services` Abschnitts der Wert ist, z. B. `services = <name-of-services-section>`

Der `services` Abschnitt ist weiter durch `<SERVICE> =` Zeilen in Unterabschnitte unterteilt, wobei sich der `<SERVICE>` AWS-Service Identifikationsschlüssel befindet. Der AWS-Service Bezeichner basiert auf dem API-Modell, indem alle Leerzeichen `serviceId` durch Unterstriche ersetzt und alle Buchstaben klein geschrieben werden. Eine Liste aller Service-ID-Schlüssel, die im `services`-Abschnitt verwendet werden können, finden Sie unter Identifikatoren für dienstspezifische Endpunkte. Auf den Service-ID-Schlüssel folgen verschachtelte Einstellungen, die jeweils in einer eigenen Zeile stehen, welche durch zwei Leerzeichen eingerückt sind.

Das folgende Beispiel verwendet eine `services` Definition, um den Endpunkt so zu konfigurieren, dass er nur für Anfragen verwendet wird, die an den Amazon DynamoDB Dienst gestellt werden.

Der "local-dynamodb" services Abschnitt wird im profile Abschnitt unter Verwendung des services Schlüssels namentlich referenziert. Der AWS-Service Identifikationsschlüssel lautetdynamodb. Der Unterabschnitt Amazon DynamoDB Service beginnt in der Zeiledynamodb = . Alle unmittelbar folgenden Zeilen, die eingerückt sind, sind in diesem Unterabschnitt enthalten und gelten für diesen Service.

```
[profile dev]
services = local-dynamodb

[services local-dynamodb]
dynamodb =
  endpoint_url = http://localhost:8000
```

Weitere Informationen zur Konfiguration benutzerdefinierter Endgeräte finden Sie unterService-specific Endpunkte.

Format der Datei mit den Anmeldeinformationen

Die Regeln für die credentials Datei sind im Allgemeinen identisch mit denen für die config Datei, mit der Ausnahme, dass Profilabschnitte nicht mit dem Wort beginnenprofile. Verwenden Sie nur den Profilnamen selbst in eckigen Klammern. Das folgende Beispiel zeigt eine credentials Datei mit einem benannten Profilabschnitt namensfoo.

```
[foo]
...credential settings...
```

Nur die folgenden Einstellungen, die als „geheim“ oder vertraulich gelten, können in der credentials Datei gespeichert werden: aws_access_key_idaws_secret_access_key, undaws_session_token. Diese Einstellungen können zwar auch in der gemeinsam genutzten config Datei platziert werden, wir empfehlen jedoch, diese sensiblen Werte in der separaten credentials Datei beizubehalten. Auf diese Weise können Sie bei Bedarf separate Berechtigungen für jede Datei bereitstellen.

Das folgende Beispiel zeigt eine credentials Basisdatei mit einem [default] Profil. Es legt die aws_session_token globalen Einstellungen für aws_access_key_idaws_secret_access_key, und fest.

```
[default]
aws_access_key_id=AKIAIOSFODNN7EXAMPLE
```

```
aws_secret_access_key=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
aws_session_token=IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZVERYLONGSTRI
```

Unabhängig davon, ob Sie ein benanntes Profil oder `default` "" in Ihrer `credentials` Datei verwenden, werden alle Einstellungen hier mit allen Einstellungen aus Ihrer `config` Datei kombiniert, die denselben Profilnamen verwendet. Wenn in beiden Dateien Anmeldeinformationen für ein Profil mit demselben Namen vorhanden sind, haben die Schlüssel in der Anmeldeinformationsdatei Vorrang.

Suchen und Ändern des Speicherorts der geteilten **`credentials`** Dateien **`config`** AWS SDKs und Tools

Bei den gemeinsam genutzten `credentials` Dateien AWS `config` und Dateien handelt es sich um Klartextdateien, die Konfigurationsinformationen für die Tools AWS SDKs und enthalten. Die Dateien befinden sich lokal in Ihrer Umgebung und werden automatisch vom SDK-Code oder von AWS CLI Befehlen verwendet, die Sie in dieser Umgebung ausführen. Zum Beispiel auf Ihrem eigenen Computer oder bei der Entwicklung auf einer Amazon Elastic Compute Cloud-Instanz.

Wenn das SDK oder Tool ausgeführt wird, sucht es nach diesen Dateien und lädt alle verfügbaren Konfigurationseinstellungen. Wenn die Dateien noch nicht vorhanden sind, wird vom SDK oder Tool automatisch eine Basisdatei erstellt.

Standardmäßig befinden sich die Dateien in einem Ordner mit dem Namen `.aws`, der sich in Ihrem Ordner `home` oder Ihrem Benutzerordner befindet.

Betriebssystem	Standardspeicherort und Name der Dateien

Unter Linux und macOS	`~/.aws/config`
	
	`~/.aws/credentials`

Windows	`%USERPROFILE%\.aws\config`
	
	`%USERPROFILE%\.aws\credentials`

Auflösung des Home-Verzeichnisses

~wird nur für die Auflösung des Home-Verzeichnisses verwendet, wenn

- Startet den Pfad
- Darauf folgt unmittelbar ein plattformspezifisches Trennzeichen `/` oder ein plattformspezifisches Trennzeichen. Unter Windows, `~/` und `~\` beide werden in das Home-Verzeichnis aufgelöst.

Bei der Bestimmung des Home-Verzeichnisses werden die folgenden Variablen geprüft:

- (Alle Plattformen) Die `HOME` Umgebungsvariable
- (Windows-Plattformen) Die `USERPROFILE` Umgebungsvariable
- (Windows-Plattformen) Die Verkettung von Variablen `HOMEDRIVE` und `HOMEPATH` Umgebungsvariablen () `$HOMEDRIVE$HOMEPATH`
- (Optional pro SDK oder Tool) Eine SDK- oder toolspezifische Funktion oder Variable zur Auflösung von Startpfaden

Wenn das Home-Verzeichnis eines Benutzers am Anfang des Pfads angegeben wird (z. B.`~username/`), wird es nach Möglichkeit in das Home-Verzeichnis des angeforderten Benutzernamens aufgelöst (z. B.`/home/username/.aws/config`).

Ändern Sie den Standardspeicherort dieser Dateien

Sie können eine der folgenden Optionen verwenden, um zu ändern, woher diese Dateien vom SDK oder Tool geladen werden.

Verwenden Sie Umgebungsvariablen

Die folgenden Umgebungsvariablen können festgelegt werden, um den Speicherort oder den Namen dieser Dateien vom Standardwert in einen benutzerdefinierten Wert zu ändern:

- `config`Datei-Umgebungsvariable: **`AWS_CONFIG_FILE`**
- `credentials`Dateiumgebungsvariable: **`AWS_SHARED_CREDENTIALS_FILE`**

Linux/macOS

Sie können einen alternativen Speicherort angeben, indem Sie die folgenden Exportbefehle unter Linux oder macOS ausführen.

```
$ export AWS_CONFIG_FILE=/some/file/path/on/the/system/config-file-name
```

```
$ export AWS_SHARED_CREDENTIALS_FILE=/some/other/file/path/on/the/system/
credentials-file-name
```

Windows

Sie können einen alternativen Speicherort angeben, indem Sie die folgenden setx-Befehle unter Windows ausführen.

```
C:\> setx AWS_CONFIG_FILE c:\some\file\path\on\the\system\config-file-name
C:\> setx AWS_SHARED_CREDENTIALS_FILE c:\some\other\file\path\on\the\system
\credentials-file-name
```

Weitere Informationen zur Konfiguration Ihres Systems mithilfe von Umgebungsvariablen finden Sie unterVerwendung von Umgebungsvariablen zur globalen Konfiguration AWS SDKs und Tools.

Verwenden Sie JVM-Systemeigenschaften

Für das SDK für Kotlin, das auf der JVM läuft, und für das SDK for Java 2.x können Sie die folgenden JVM-Systemeigenschaften festlegen, um den Speicherort oder den Namen dieser Dateien vom Standard auf einen benutzerdefinierten Wert zu ändern:

- `config`Datei-JVM-Systemeigenschaft: **`aws.configFile`**
- `credentials`Datei-Umgebungsvariable: **`aws.sharedCredentialsFile`**

Anweisungen zum Einstellen der JVM-Systemeigenschaften finden Sie unterthe section called “Wie legt man die JVM-Systemeigenschaften fest”. Das SDK for Java 1.x unterstützt diese Systemeigenschaften nicht.

Verwendung von Umgebungsvariablen zur globalen Konfiguration AWS SDKs und Tools

Umgebungsvariablen bieten eine weitere Möglichkeit, Konfigurationsoptionen und Anmeldeinformationen bei der Verwendung von AWS SDKs AND-Tools anzugeben. Umgebungsvariablen können nützlich sein, um Skripts zu erstellen oder vorübergehend ein benanntes Profil als Standard festzulegen. Eine Liste der Umgebungsvariablen, die von den meisten unterstützt werden SDKs, finden Sie unterListe der Umgebungsvariablen.

Vorrang von Optionen

- Wenn Sie eine Einstellung mithilfe der zugehörigen Umgebungsvariablen angeben, überschreibt sie alle Werte, die aus einem Profil in den gemeinsam genutzten AWS `config` `credentials` Dateien geladen wurden.
- Wenn Sie eine Einstellung mithilfe eines Parameters in der AWS CLI Befehlszeile angeben, überschreibt sie jeden Wert aus der entsprechenden Umgebungsvariablen oder einem Profil in der Konfigurationsdatei.

Festlegen von Umgebungsvariablen

Die folgenden Beispiele zeigen, wie Sie Umgebungsvariablen für den Standardbenutzer konfigurieren können.

Linux, macOS, or Unix

```
$ export AWS_ACCESS_KEY_ID=AKIAIOSFODNN7EXAMPLE
$ export AWS_SECRET_ACCESS_KEY=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
$ export
 AWS_SESSION_TOKEN=AQoEXAMPLEH4aoAH0gNCAPy...truncated...zrkuWJOgQs8IZZaIv2BXIa2R4Olgk
$ export AWS_REGION=us-west-2
```

Durch die Festlegung der Umgebungsvariablen wird der verwendete Wert bis zum Ende der Shell-Sitzung oder bis zur Festlegung eines anderen Wertes für die Variable geändert. Sie können Variablen für zukünftige Sitzungen persistent machen, indem Sie sie im Startup-Skript Ihrer Shell festlegen.

Windows Command Prompt

```
C:\> setx AWS_ACCESS_KEY_ID AKIAIOSFODNN7EXAMPLE
C:\> setx AWS_SECRET_ACCESS_KEY wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
C:\> setx
 AWS_SESSION_TOKEN AQoEXAMPLEH4aoAH0gNCAPy...truncated...zrkuWJOgQs8IZZaIv2BXIa2R4Olgk
C:\> setx AWS_REGION us-west-2
```

Wenn `set` Sie eine Umgebungsvariable festlegen, ändert sich der verwendete Wert bis zum Ende der aktuellen Befehlszeilensitzung oder bis Sie die Variable auf einen anderen Wert setzen. Wenn `setx`Sie eine Umgebungsvariable festlegen, ändert sich der Wert, der sowohl in der aktuellen Eingabeaufforderungssitzung als auch in allen Befehlszeilensitzungen verwendet wird,

die Sie nach der Ausführung des Befehls erstellen. Andere Befehls-Shells, die zum Zeitpunkt der Befehlsausführung bereits ausgeführt werden, sind hiervon nicht betroffen.

PowerShell

```
PS C:\> $Env:AWS_ACCESS_KEY_ID="AKIAIOSFODNN7EXAMPLE"
PS C:\> $Env:AWS_SECRET_ACCESS_KEY="wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY"
PS C:
\> $Env:AWS_SESSION_TOKEN="AQoEXAMPLEH4aoAH0gNCAPy...truncated...zrkuWJOgQs8IZZaIv2BXIa2R4Ol
PS C:\> $Env:AWS_REGION="us-west-2"
```

Wenn Sie an der PowerShell Eingabeaufforderung eine Umgebungsvariable festlegen, wie in den vorherigen Beispielen gezeigt, wird der Wert nur für die Dauer der aktuellen Sitzung gespeichert. Um die Einstellung der Umgebungsvariablen für alle Sitzungen PowerShell und Befehlszeilensitzungen beizubehalten, speichern Sie sie mithilfe der Systemanwendung in der Systemsteuerung. Alternativ können Sie die Variable für alle future PowerShell Sitzungen festlegen, indem Sie sie Ihrem PowerShell Profil hinzufügen. Weitere Informationen zum Speichern von Umgebungsvariablen oder deren Beibehaltung über mehrere Sitzungen hinweg finden Sie in der PowerShell Dokumentation.

Einrichtung von serverlosen Umgebungsvariablen

Wenn Sie eine serverlose Architektur für die Entwicklung verwenden, haben Sie andere Optionen zum Setzen von Umgebungsvariablen. Abhängig von Ihrem Container können Sie unterschiedliche Strategien für Code verwenden, der in diesen Containern ausgeführt wird, um Umgebungsvariablen zu sehen und darauf zuzugreifen, ähnlich wie in Nicht-Cloud-Umgebungen.

Mit können Sie AWS Lambda beispielsweise Umgebungsvariablen direkt festlegen. Einzelheiten finden Sie unter Verwenden von AWS Lambda Umgebungsvariablen im AWS Lambda Entwicklerhandbuch.

In Serverless Framework können Sie häufig SDK-Umgebungsvariablen in der `serverless.yml` Datei unter dem Provider-Schlüssel unter der Umgebungseinstellung festlegen. Informationen zur `serverless.yml` Datei finden Sie unter Allgemeine Funktionseinstellungen in der Serverless Framework-Dokumentation.

Unabhängig davon, welchen Mechanismus Sie zum Setzen von Container-Umgebungsvariablen verwenden, gibt es einige, die vom Container reserviert sind, z. B. diejenigen, die für Lambda at Defined Runtime-Umgebungsvariablen dokumentiert sind. Schlagen Sie immer in der offiziellen

Dokumentation des Containers nach, den Sie verwenden, um festzustellen, wie Umgebungsvariablen behandelt werden und ob es Einschränkungen gibt.

Verwenden von JVM-Systemeigenschaften zur globalen Konfiguration AWS SDK für Java und AWS SDK für Kotlin

JVM-Systemeigenschaften bieten eine weitere Möglichkeit, Konfigurationsoptionen und Anmeldeinformationen für Dateien anzugeben SDKs , die auf der JVM ausgeführt werden, z. B. die und die AWS SDK für Java . AWS SDK für KotlinEine Liste der JVM-Systemeigenschaften, die von unterstützt werden SDKs, finden Sie in der Einstellungsreferenz.

Vorrang von Optionen

- Wenn Sie eine Einstellung mithilfe ihrer JVM-Systemeigenschaft angeben, überschreibt sie jeden Wert, der in Umgebungsvariablen gefunden oder aus einem Profil in den gemeinsam genutzten AWS `config` und `credentials` Dateien geladen wurde.
- Wenn Sie eine Einstellung mithilfe der zugehörigen Umgebungsvariablen angeben, überschreibt sie alle Werte, die aus einem Profil im gemeinsam genutzten AWS `config` und in den `credentials` Dateien geladen wurden.

Wie legt man die JVM-Systemeigenschaften fest

Sie können die JVM-Systemeigenschaften auf verschiedene Arten festlegen.

In der Befehlszeile

Stellen Sie die JVM-Systemeigenschaften in der Befehlszeile ein, wenn Sie den `java` Befehl mit dem Switch aufrufen. `-D` Der folgende Befehl konfiguriert AWS-Region global für alle Service-Clients, sofern Sie den Wert im Code nicht explizit überschreiben.

```
java -Daws.region=us-east-1 -jar <your_application.jar> <other_arguments>
```

Wenn Sie mehrere JVM-Systemeigenschaften festlegen müssen, geben Sie den `-D` Switch mehrmals an.

Mit einer Umgebungsvariablen

Wenn Sie nicht auf die Befehlszeile zugreifen können, um die JVM zum Ausführen Ihrer Anwendung aufzurufen, können Sie die `JAVA_TOOL_OPTIONS` Umgebungsvariable verwenden, um Befehlszeilenoptionen zu konfigurieren. Dieser Ansatz ist beispielsweise beim Ausführen einer AWS Lambda Funktion in der Java-Laufzeit oder beim Ausführen von Code in einer eingebetteten JVM nützlich.

Im folgenden Beispiel wird AWS-Region global für alle Service-Clients konfiguriert, sofern Sie den Wert im Code nicht explizit überschreiben.

Linux, macOS, or Unix

```
$ export JAVA_TOOL_OPTIONS="-Daws.region=us-east-1"
```

Durch die Festlegung der Umgebungsvariablen wird der verwendete Wert bis zum Ende der Shell-Sitzung oder bis zur Festlegung eines anderen Wertes für die Variable geändert. Sie können Variablen für zukünftige Sitzungen persistent machen, indem Sie sie im Startup-Skript Ihrer Shell festlegen.

Windows Command Prompt

```
C:\> setx JAVA_TOOL_OPTIONS -Daws.region=us-east-1
```

Wenn `set` Sie eine Umgebungsvariable festlegen, ändert sich der verwendete Wert bis zum Ende der aktuellen Eingabeaufforderungssitzung oder bis Sie die Variable auf einen anderen Wert setzen. Wenn `setx`Sie eine Umgebungsvariable festlegen, ändert sich der Wert, der sowohl in der aktuellen Eingabeaufforderungssitzung als auch in allen Befehlszeilensitzungen verwendet wird, die Sie nach der Ausführung des Befehls erstellen. Andere Befehls-Shells, die zum Zeitpunkt der Befehlsausführung bereits ausgeführt werden, sind hiervon nicht betroffen.

Zur Laufzeit

Sie können JVM-Systemeigenschaften auch zur Laufzeit im Code festlegen, indem Sie die `System.setProperty` Methode verwenden, wie im folgenden Beispiel gezeigt.

```
System.setProperty("aws.region", "us-east-1");
```

Important

Legen Sie alle JVM-Systemeigenschaften fest, bevor Sie SDK-Dienstclients initialisieren, da Dienstclients andernfalls möglicherweise andere Werte verwenden.

Authentifizierung und Zugriff mithilfe von AWS SDKs Tools

Wenn Sie eine AWS SDK-Anwendung entwickeln oder zu verwendende AWS Tools verwenden AWS-Services, müssen Sie festlegen, mit welcher Methode Ihr Code oder Tool authentifiziert wird. AWS Sie können den programmatischen Zugriff auf AWS Ressourcen auf unterschiedliche Weise konfigurieren, abhängig von der Umgebung, in der der Code ausgeführt wird, und dem verfügbaren AWS Zugriff.

Die folgenden Optionen sind Teil der Anmeldeinformationsanbieterkette. Das bedeutet, dass Ihr AWS SDK oder Tool diese Authentifizierungsmethode automatisch erkennt AWS `config` und verwendet, wenn Sie Ihre geteilten `credentials` Dateien und Dateien entsprechend konfigurieren.

Wählen Sie eine Methode zur Authentifizierung Ihres Anwendungscodes

Wählen Sie eine Methode zur Authentifizierung der Aufrufe, an die Ihre Anwendung gesendet AWS hat.

Führen Sie Code INSIDE aus AWS-Service (wie Amazon EC2, Lambda, Amazon ECS, Amazon EKS, CodeBuild)?

Wenn Ihr Code auf läuft AWS, können Anmeldeinformationen automatisch für Ihre Anwendung verfügbar gemacht werden. Wenn Ihre Anwendung beispielsweise auf Amazon Elastic Compute Cloud gehostet wird und dieser Ressource eine IAM-Rolle zugeordnet ist, werden die Anmeldeinformationen automatisch für Ihre Anwendung verfügbar gemacht. Wenn Sie Amazon ECS- oder Amazon EKS-Container verwenden, können die für die IAM-Rolle festgelegten Anmeldeinformationen ebenfalls automatisch abgerufen werden, indem der Code innerhalb des Containers über die Anmeldeinformationsanbieterkette des SDK ausgeführt wird.

Befindet sich Ihr Code in einer Amazon Elastic Compute Cloud-Instanz?

Verwenden von IAM-Rollen zur Authentifizierung von Anwendungen, die auf Amazon bereitgestellt werden EC2— Verwenden Sie IAM-Rollen, um Ihre Anwendung sicher auf einer Amazon EC2 EC2-Instance auszuführen.

Ist Ihr Code in einer Funktion? AWS Lambda

Lambda erstellt eine Ausführungsrolle mit minimalen Berechtigungen, wenn Sie eine Lambda-Funktion erstellen. Das AWS SDK oder Tool verwendet dann automatisch die IAM-Rolle, die dem Lambda zur Laufzeit über die Lambda-Ausführungsumgebung zugewiesen ist.

Befindet sich Ihr Code in Amazon Elastic Container Service (auf Amazon EC2 oder AWS Fargate für Amazon ECS)?

Verwenden Sie die IAM-Rolle für die Aufgabe. Sie müssen eine Aufgabenrolle erstellen und diese Rolle in Ihrer Amazon ECS-Aufgabendefinition angeben. Das AWS SDK oder Tool verwendet dann automatisch die der Aufgabe zur Laufzeit zugewiesene IAM-Rolle über die Amazon ECS-Metadaten.

Ist Ihr Code in Amazon Elastic Kubernetes Service?

Wir empfehlen Ihnen, Amazon EKS Pod Identities zu verwenden.

Hinweis: Wenn Sie der Meinung sind, dass IAM-Rollen für Service Accounts (IRSA) besser auf Ihre individuellen Bedürfnisse zugeschnitten sind, finden Sie weitere Informationen unter Vergleich von EKS Pod Identity und IRSA im Amazon EKS-Benutzerhandbuch.

Läuft Ihr Code in AWS CodeBuild

Weitere Informationen finden Sie unter Verwenden identitätsbasierter Richtlinien für. CodeBuild

Ist Ihr Code in einem anderen? AWS-Service

Sehen Sie sich den speziellen Leitfaden für Sie an AWS-Service. Wenn Sie Code on ausführen AWS, kann die SDK-Anmeldeinformationsanbieterkette automatisch Anmeldeinformationen für Sie abrufen und aktualisieren.

Erstellen Sie mobile Anwendungen oder clientbasierte Webanwendungen?

Wenn Sie mobile Anwendungen oder clientbasierte Webanwendungen erstellen, auf die Zugriff erforderlich ist AWS, erstellen Sie Ihre App so, dass sie mithilfe eines Web-Identitätsverbunds dynamisch temporäre AWS Sicherheitsanmeldeinformationen anfordert.

Mit Web-Identitätsverbund müssen Sie keinen eigenen Anmeldecode schreiben oder eigene Benutzeridentitäten verwalten. Stattdessen können sich App-Nutzer mit einem bekannten externen Identitätsanbieter (IdP) anmelden, z. B. Login with Amazon, Facebook, Google oder einem anderen OpenID Connect (OIDC) -kompatiblen IdP. Sie können ein Authentifizierungstoken erhalten und

dieses Token dann gegen temporäre Sicherheitsanmeldeinformationen in AWS dieser Zuordnung zu einer IAM-Rolle mit Berechtigungen zur Nutzung der Ressourcen in Ihrem eintauschen. AWS-Konto

Wie Sie dies für Ihr SDK oder Tool konfigurieren, erfahren Sie unterÜbernahme einer Rolle bei Web Identity oder OpenID Connect zur Authentifizierung und Tools AWS SDKs.

Erwägen Sie für mobile Anwendungen die Verwendung von Amazon Cognito. Amazon Cognito fungiert als Identitätsbroker und erledigt einen Großteil der Verbundarbeit für Sie. Weitere Informationen finden Sie unter Verwenden von Amazon Cognito für mobile Apps im IAM-Benutzerhandbuch.

Entwickeln und führen Sie den Code LOKAL aus?

Wir empfehlenVerwendung von Konsolenanmeldedaten zur Authentifizierung AWS SDKs und Tools.

Nach einem schnellen browserbasierten Authentifizierungsablauf AWS werden automatisch temporäre Anmeldeinformationen generiert, die in allen lokalen Entwicklungstools wie der AWS CLI AWS -Tools für PowerShell und AWS SDKs funktionieren.

Wenn Sie Identity Center für den AWS Kontozugriff verwenden

Verwenden Sie IAM Identity Center, um AWS SDK und Tools zu authentifizieren, wenn Sie bereits Zugriff auf AWS Konten haben, die den Zugriff für Ihre Belegschaft verwalten and/or müssen. Aus Sicherheitsgründen empfehlen wir die Verwendung AWS Organizations zusammen mit IAM Identity Center, um den Zugriff für alle Ihre Konten zu verwalten. AWS Sie können Benutzer im IAM Identity Center erstellen, Microsoft Active Directory verwenden, einen SAML 2.0-Identitätsanbieter (IdP) verwenden oder Ihren IdP individuell mit Konten verbinden. AWS Um zu überprüfen, ob Ihre Region IAM Identity Center unterstützt, finden Sie unter Verwenden von IAM Identity Center zur Authentifizierung von AWS SDK und Tools IAM Identity Center-Endpunkte und Kontingente in der Amazon Web Services General Reference.

Wenn Sie nach anderen Authentifizierungsmöglichkeiten suchen

Erstellen Sie einen IAM-Benutzer mit den geringsten Rechten und Berechtigungen für den Zugriff auf `sts:AssumeRole` Ihre Zielrolle. Konfigurieren Sie dann Ihr Profil so, dass es eine Rolle annimmt, indem Sie eine für diesen Benutzer `source_profile` eingerichtete Konfiguration verwenden.

Sie können temporäre IAM-Anmeldeinformationen auch über Umgebungsvariablen oder die Datei mit gemeinsam genutzten AWS Anmeldeinformationen verwenden. Weitere Informationen finden Sie unter Verwenden von kurzfristigen Anmeldeinformationen zur Authentifizierung AWS SDKs und zu Tools.

Hinweis: Nur in Sandbox- oder Lernumgebungen können Sie erwägen, langfristige Anmeldeinformationen für die Authentifizierung AWS SDKs und Tools zu verwenden.

Wird dieser Code vor Ort oder in einer Hybrid-/On-Demand-VM ausgeführt (z. B. auf einem Server, der von Amazon S3 liest oder auf Amazon S3 schreibt, oder Jenkins, der in der Cloud bereitstellt)?

Verwenden Sie X.509-Client-Zertifikate?

Ja: SieheAuthentifizierung AWS SDKs und Tools mit IAM Roles Anywhere. Sie können IAM Roles Anywhere verwenden, um temporäre Sicherheitsanmeldeinformationen in IAM für Workloads wie Server, Container und Anwendungen abzurufen, die außerhalb von ausgeführt werden. AWS Um IAM Roles Anywhere verwenden zu können, müssen Ihre Workloads X.509-Zertifikate verwenden.

Kann die Umgebung eine sichere Verbindung zu einem Federated Identity Provider (wie Microsoft Entra oder Okta) herstellen, um temporäre Anmeldeinformationen anzufordern? AWS

Ja: Verwenden Anbieter von Prozessanmeldeinformationen

Wird verwendetAnbieter von Prozessanmeldeinformationen, um Anmeldeinformationen zur Laufzeit automatisch abzurufen. Diese Systeme verwenden möglicherweise ein Hilfstool oder ein Plug-in, um die Anmeldeinformationen abzurufen, und übernehmen möglicherweise im Hintergrund eine IAM-Rolle mithilfe von`sts:AssumeRole`.

Nein: Verwenden Sie temporäre Anmeldeinformationen, die über eingegeben wurden AWS Secrets Manager

Verwenden Sie temporäre Anmeldeinformationen, die über eingegeben wurden AWS Secrets Manager. Optionen zum Abrufen kurzlebiger Zugriffsschlüssel finden Sie unter Temporäre Sicherheitsanmeldedaten anfordern im IAM-Benutzerhandbuch. Optionen zum Speichern dieser temporären Anmeldeinformationen finden Sie unter. AWS Zugriffstasten

Sie können diese Anmeldeinformationen verwenden, um umfassendere Anwendungsberechtigungen sicher aus Secrets Manager abzurufen, wo Ihre Produktionsgeheimnisse oder langlebige rollenbasierte Anmeldeinformationen gespeichert werden können.

Verwenden Sie ein Drittanbieter-Tool, das nicht enthalten ist? AWS

Die beste Anleitung zur Beschaffung von Anmeldeinformationen finden Sie in der von Ihrem Drittanbieter verfassten Dokumentation.

Können Sie temporäre Anmeldeinformationen sicher eingeben, wenn Ihr Drittanbieter keine Unterlagen bereitgestellt hat?

Ja: Verwenden Sie Umgebungsvariablen und temporäre AWS STS Anmeldeinformationen.

Nein: Verwenden Sie statische Zugriffsschlüssel, die im verschlüsselten Secret Manager gespeichert sind (letzter Ausweg).

Authentifizierungsmethoden

Authentifizierungsmethoden für Code, der in einer AWS Umgebung ausgeführt wird

Wenn Ihr Code auf läuft AWS, können Anmeldeinformationen automatisch für Ihre Anwendung verfügbar gemacht werden. Wenn Ihre Anwendung beispielsweise auf Amazon Elastic Compute Cloud gehostet wird und dieser Ressource eine IAM-Rolle zugeordnet ist, werden die Anmeldeinformationen automatisch für Ihre Anwendung verfügbar gemacht. Wenn Sie Amazon ECS- oder Amazon EKS-Container verwenden, können die für die IAM-Rolle festgelegten Anmeldeinformationen ebenfalls automatisch abgerufen werden, indem der Code innerhalb des Containers über die Anmeldeinformationsanbieterkette des SDK ausgeführt wird.

- Verwenden von IAM-Rollen zur Authentifizierung von Anwendungen, die auf Amazon bereitgestellt werden EC2— Verwenden Sie IAM-Rollen, um Ihre Anwendung sicher auf einer Amazon EC2 EC2-Instance auszuführen.
- Sie können auf folgende Weise programmgesteuert mit der AWS Nutzung von IAM Identity Center interagieren:
  - Wird verwendet AWS CloudShell, um AWS CLI Befehle von der Konsole aus auszuführen.
  - Wenn Sie einen cloudbasierten Kollaborationsraum für Softwareentwicklungsteams ausprobieren möchten, sollten Sie Amazon in Betracht ziehen CodeCatalyst.

Authentifizierung über einen webbasierten Identitätsanbieter — mobile oder clientbasierte Webanwendungen

Wenn Sie mobile Anwendungen oder clientbasierte Webanwendungen erstellen, auf die Zugriff erforderlich ist AWS, erstellen Sie Ihre App so, dass sie mithilfe eines Web-Identitätsverbunds dynamisch temporäre AWS Sicherheitsanmeldeinformationen anfordert.

Mit Web-Identitätsverbund müssen Sie keinen eigenen Anmeldecode schreiben oder eigene Benutzeridentitäten verwalten. Stattdessen können sich App-Nutzer mit einem bekannten externen

Identitätsanbieter (IdP) anmelden, z. B. Login with Amazon, Facebook, Google oder einem anderen OpenID Connect (OIDC) -kompatiblen IdP. Sie können ein Authentifizierungstoken erhalten und dieses Token dann gegen temporäre Sicherheitsanmeldeinformationen in AWS dieser Zuordnung zu einer IAM-Rolle mit Berechtigungen zur Nutzung der Ressourcen in Ihrem eintauschen. AWS-Konto

Wie Sie dies für Ihr SDK oder Tool konfigurieren, erfahren Sie unterÜbernahme einer Rolle bei Web Identity oder OpenID Connect zur Authentifizierung und Tools AWS SDKs.

Erwägen Sie für mobile Anwendungen die Verwendung von Amazon Cognito. Amazon Cognito fungiert als Identitätsbroker und erledigt einen Großteil der Verbundarbeit für Sie. Weitere Informationen finden Sie unter Verwenden von Amazon Cognito für mobile Apps im IAM-Benutzerhandbuch.

Authentifizierungsmethoden für Code, der lokal ausgeführt wird (nicht in) AWS

- Verwendung von Konsolenanmeldedaten zur Authentifizierung AWS SDKs und Tools— Diese Funktion funktioniert sowohl mit der AWS Befehlszeilenschnittstelle als auch mit Tools für PowerShell und bietet Ihnen aktualisierbare Anmeldeinformationen, die für alle lokalen Entwicklungstools wie AWS CLI, Tools for PowerShell und AWS verwendet werden können.
- Verwenden von IAM Identity Center zur Authentifizierung von AWS SDK und Tools— Aus Sicherheitsgründen empfehlen wir die Verwendung AWS Organizations zusammen mit IAM Identity Center, um den Zugriff für alle Ihre Benutzer zu verwalten. AWS-Konten Sie können Benutzer in Microsoft Active Directory erstellen AWS IAM Identity Center, einen SAML 2.0-Identitätsanbieter (IdP) verwenden oder Ihren IdP individuell mit diesem verbinden. AWS-Konten Informationen darüber, ob Ihre Region IAM Identity Center unterstützt, finden Sie unter AWS IAM Identity Center Endpunkte und Kontingente in der. Allgemeine Amazon Web Services-Referenz
- Authentifizierung AWS SDKs und Tools mit IAM Roles Anywhere— Sie können IAM Roles Anywhere verwenden, um temporäre Sicherheitsanmeldedaten in IAM für Workloads wie Server, Container und Anwendungen abzurufen, die außerhalb von ausgeführt werden. AWS Um IAM Roles Anywhere verwenden zu können, müssen Ihre Workloads X.509-Zertifikate verwenden.
- Übernahme einer Rolle mit AWS Anmeldeinformationen zur Authentifizierung AWS SDKs und Tools— Sie können eine IAM-Rolle annehmen, um vorübergehend auf AWS Ressourcen zuzugreifen, auf die Sie sonst möglicherweise keinen Zugriff hätten.
- Verwendung von AWS Zugriffsschlüsseln zur Authentifizierung AWS SDKs und Tools— Andere Optionen, die möglicherweise weniger praktisch sind oder das Sicherheitsrisiko für Ihre AWS Ressourcen erhöhen könnten.

Weitere Informationen zur Zugriffsverwaltung

Das IAM-Benutzerhandbuch enthält die folgenden Informationen zur sicheren Steuerung des Zugriffs auf AWS Ressourcen:

- IAM-Identitäten (Benutzer, Benutzergruppen und Rollen) — Verstehen Sie die Grundlagen von Identitäten in. AWS
- Bewährte Sicherheitspraktiken in IAM — Sicherheitsempfehlungen, die bei der Entwicklung von AWS Anwendungen nach dem Modell der geteilten Verantwortung zu beachten sind.

Das Allgemeine Amazon Web Services-Referenzenthält grundlegende Grundlagen zu den folgenden Themen:

- Ihre AWS Anmeldeinformationen verstehen und abrufen — Zugriff auf wichtige Optionen und Verwaltungspraktiken sowohl für den Konsolen- als auch für den programmgesteuerten Zugriff.

Zugriff auf das IAM Identity Center-Plugin für Trusted Identity Propagation (TIP) AWS-Services

- Verwenden des TIP-Plugins für den Zugriff AWS-Services— Wenn Sie eine Anwendung für Amazon Q Business oder einen anderen Service erstellen, der die Verbreitung vertrauenswürdiger Identitäten unterstützt, und das AWS SDK für Java oder das verwenden AWS SDK für JavaScript, können Sie das TIP-Plugin verwenden, um die Autorisierung zu vereinfachen.

AWS Builder ID

Sie AWS Builder ID ergänzen alle, die AWS-Konten Sie vielleicht schon besitzen oder erstellen möchten. Eine AWS-Konto fungiert zwar als Container für AWS Ressourcen, die Sie erstellen, und bietet eine Sicherheitsgrenze für diese Ressourcen, aber Ihre AWS Builder ID repräsentiert Sie als Einzelperson. Sie können sich mit Ihrem anmelden AWS Builder ID , um auf Entwicklertools und -dienste wie Amazon Q und Amazon zuzugreifen CodeCatalyst.

- Melden Sie sich AWS Builder ID im AWS-Anmeldung Benutzerhandbuch an — Erfahren Sie, wie Sie eine erstellen und verwenden, AWS Builder ID und erfahren Sie, was die Builder-ID bietet.
- CodeCatalystKonzepte — AWS Builder ID im CodeCatalyst Amazon-Benutzerhandbuch — Erfahren Sie, wie ein CodeCatalyst verwendet wird AWS Builder ID.

Verwendung von Konsolenanmeldedaten zur Authentifizierung AWS SDKs und Tools

Die Verwendung von Konsolenanmeldedaten ist die empfohlene Methode zur Bereitstellung von AWS Anmeldeinformationen, wenn Sie eine AWS Anwendung in Ihrer lokalen Umgebung oder in anderen Serviceumgebungen ohne AWS Rechenleistung entwickeln. Wenn Sie auf einer AWS Ressource wie Amazon Elastic Compute Cloud (Amazon EC2) oder entwickeln, empfehlen wir AWS CloudShell, stattdessen Anmeldeinformationen von diesem Service zu beziehen.

Sie können sich auch über das IAM Identity Center authentifizieren. Verwenden von IAM Identity Center zur Authentifizierung von AWS SDK und Tools Diese Option ist eine gängige Methode für Unternehmen, um den Zugriff für ihre Belegschaft zu verwalten, und setzt voraus, dass Identity Center aktiviert ist.

Funktionsweise

Wenn Sie sich mit Konsolenanmeldedaten für die AWS lokale Entwicklung anmelden, können Sie Ihre vorhandenen Anmeldedaten für die AWS Management Console für den programmatischen Zugriff AWS auf Dienste verwenden. AWS Generiert nach einem browserbasierten Authentifizierungsablauf temporäre Anmeldeinformationen, die in allen lokalen Entwicklungstools wie AWS CLI, Tools for PowerShell und AWS SDKs funktionieren. Diese Funktion vereinfacht die Konfiguration und Verwaltung von AWS CLI-Anmeldeinformationen, insbesondere wenn Sie die interaktive Authentifizierung der Verwaltung von langfristigen Zugriffsschlüsseln vorziehen.

Mit diesem Prozess können Sie sich mit Ihren Root-Anmeldeinformationen, die Sie bei der ersten Kontoeinrichtung erstellt haben, mit IAM-Benutzern oder mit einer föderierten Identität von Ihrem Identitätsanbieter authentifizieren.

Wenn Sie sie SDKs für die Entwicklung verwenden, verwenden die SDK-Clients die temporären Anmeldeinformationen über die. AWS SDKs und Tools standardisierte Anbieter von Anmeldeinformationen Sie können auch die konfigurierenAnbieter von Anmeldedaten.

Die Authentifizierung über den Login-Befehl wird sowohl von AWS CLI als auch von Tools unterstützt für PowerShell:

- Melden Sie sich mit Konsolenanmeldedaten für die AWS lokale Entwicklung an
- Melden Sie sich mit den Konsolenanmeldedaten im AWS -Tools für PowerShell Benutzerhandbuch an

Verwenden von IAM Identity Center zur Authentifizierung von AWS SDK und Tools

AWS IAM Identity Center kann verwendet werden, um AWS Anmeldeinformationen bereitzustellen, wenn eine AWS Anwendung in Serviceumgebungen ohne AWS Rechenleistung entwickelt wird. Wenn Sie auf einer AWS Ressource wie Amazon Elastic Compute Cloud (Amazon EC2) oder entwickeln, empfehlen wir AWS Cloud9, stattdessen Anmeldeinformationen von diesem Service zu beziehen.

Verwenden Sie die IAM Identity Center-Authentifizierung, wenn Sie Identity Center bereits für den AWS Kontozugriff verwenden oder den Zugriff für eine Organisation verwalten müssen.

In diesem Tutorial richten Sie den IAM Identity Center-Zugriff ein und konfigurieren ihn für Ihr SDK oder Tool mithilfe des AWS Zugriffsportals und des. AWS CLI

- Das AWS Zugriffsportal ist die Webadresse, über die Sie sich manuell beim IAM Identity Center anmelden. Das Format der URL ist `d-xxxxxxxxxx.awsapps.com/start` oder*`your_subdomain`*`.awsapps.com/start`. Wenn Sie im AWS Access Portal angemeldet sind, können Sie die Rollen einsehen AWS-Konten , die für diesen Benutzer konfiguriert wurden. Bei diesem Verfahren werden die Konfigurationswerte, die Sie für den SDK/tool Authentifizierungsprozess benötigen, über das AWS Zugriffsportal abgerufen.
- Das AWS CLI wird verwendet, um Ihr SDK oder Tool so zu konfigurieren, dass es die IAM Identity Center-Authentifizierung für API-Aufrufe verwendet, die über Ihren Code getätigt werden. Dieser einmalige Vorgang aktualisiert Ihre gemeinsam genutzte AWS `config` Datei, die dann von Ihrem SDK oder Tool verwendet wird, wenn Sie Ihren Code ausführen.

Voraussetzungen

Bevor Sie mit diesem Verfahren beginnen, sollten Sie die folgenden Schritte abgeschlossen haben:

- Wenn Sie noch keine haben AWS-Konto, melden Sie sich für eine an AWS-Konto.
- Wenn Sie IAM Identity Center noch nicht aktiviert haben, aktivieren Sie IAM Identity Center, indem Sie den Anweisungen im AWS IAM Identity Center Benutzerhandbuch folgen.

Konfigurieren Sie den programmatischen Zugriff mit IAM Identity Center

Schritt 1: Richten Sie den Zugriff ein und wählen Sie den entsprechenden Berechtigungssatz aus

Wählen Sie eine der folgenden Methoden, um auf Ihre AWS Anmeldeinformationen zuzugreifen.

Ich habe keinen Zugriff über IAM Identity Center eingerichtet

1. Fügen Sie einen Benutzer hinzu und fügen Sie Administratorberechtigungen hinzu, indem Sie das Verfahren Benutzerzugriff mit dem standardmäßigen IAM Identity Center-Verzeichnis konfigurieren im AWS IAM Identity Center Benutzerhandbuch befolgen.
2. Der `AdministratorAccess` Berechtigungssatz sollte nicht für die reguläre Entwicklung verwendet werden. Stattdessen empfehlen wir, den vordefinierten `PowerUserAccess` Berechtigungssatz zu verwenden, es sei denn, Ihr Arbeitgeber hat zu diesem Zweck einen benutzerdefinierten Berechtigungssatz erstellt.

   Gehen Sie erneut wie beim Konfigurieren des Benutzerzugriffs mit dem standardmäßigen IAM Identity Center-Verzeichnis vor, diesmal jedoch:

   - Anstatt die *`Admin team`* Gruppe zu erstellen, erstellen Sie eine *`Dev team`* Gruppe und ersetzen Sie diese anschließend in den Anweisungen.
   - Sie können den vorhandenen Benutzer verwenden, der Benutzer muss jedoch der neuen *`Dev team`* Gruppe hinzugefügt werden.
   - Anstatt den *`AdministratorAccess`* Berechtigungssatz zu erstellen, erstellen Sie einen *`PowerUserAccess`* Berechtigungssatz und ersetzen Sie ihn anschließend in der Anleitung.

   Wenn Sie fertig sind, sollten Sie über Folgendes verfügen:

   - Eine `Dev team` Gruppe.
   - Ein `PowerUserAccess` angehängter Berechtigungssatz für die `Dev team` Gruppe.
   - Ihr Benutzer wurde der `Dev team` Gruppe hinzugefügt.
3. Verlassen Sie das Portal und melden Sie sich erneut an, um Ihre Optionen AWS-Konten und Optionen für `Administrator` oder zu sehen`PowerUserAccess`. Wählen Sie diese Option`PowerUserAccess`, wenn Sie mit Ihrem Tool/SDK arbeiten.

Ich habe bereits AWS über einen von meinem Arbeitgeber verwalteten Federated Identity Provider (wie Microsoft Entra oder Okta) Zugriff darauf

Melden Sie sich AWS über das Portal Ihres Identitätsanbieters an. Wenn Ihr Cloud-Administrator Ihnen `PowerUserAccess` (Entwickler-) Berechtigungen erteilt hat, sehen Sie, auf AWS-Konten welche Sie Zugriff haben, und Ihren Berechtigungssatz. Neben dem Namen Ihres Berechtigungssatzes sehen Sie Optionen für den manuellen oder programmgesteuerten Zugriff auf die Konten mithilfe dieses Berechtigungssatzes.

Benutzerdefinierte Implementierungen können zu unterschiedlichen Erfahrungen führen, z. B. zu unterschiedlichen Namen von Berechtigungssätzen. Wenn Sie sich nicht sicher sind, welchen Berechtigungssatz Sie verwenden sollen, wenden Sie sich an Ihr IT-Team.

Ich habe bereits Zugriff auf AWS das von meinem Arbeitgeber verwaltete AWS Zugangsportal

Melden Sie sich AWS über das AWS Zugangsportal an. Wenn Ihr Cloud-Administrator Ihnen `PowerUserAccess` (Entwickler-)Berechtigungen erteilt hat, sehen Sie die AWS-Konten , auf die Sie Zugriff haben, und Ihren Berechtigungssatz. Neben dem Namen Ihres Berechtigungssatzes sehen Sie Optionen für den manuellen oder programmgesteuerten Zugriff auf die Konten mithilfe dieses Berechtigungssatzes.

Ich habe bereits AWS über einen föderierten benutzerdefinierten Identitätsanbieter, der von meinem Arbeitgeber verwaltet wird, Zugriff darauf

Wenden Sie sich an Ihr IT-Team, um Hilfe zu erhalten.

Schritt 2: Konfiguration SDKs und Tools zur Verwendung von IAM Identity Center

1. Installieren Sie auf Ihrem Entwicklungscomputer die neueste Version AWS CLI.

    a. Weitere Informationen finden Sie im AWS Command Line Interface Benutzerhandbuch unter Installation oder Aktualisierung AWS CLI der neuesten Version von.

    b. (Optional) Um zu überprüfen, ob der AWS CLI funktioniert, öffnen Sie eine Befehlszeile und führen Sie den `aws --version` Befehl aus.

2. Melden Sie sich beim AWS Access Portal an. Ihr Arbeitgeber kann diese URL angeben oder Sie erhalten sie in einer E-Mail nach Schritt 1: Zugriff einrichten. Wenn nicht, suchen Sie die URL Ihres AWS Zugangsportals im Dashboard von https://console.aws.amazon.com/singlesignon/.

    a. Wählen Sie im AWS Zugriffsportal auf der Registerkarte Konten das einzelne Konto aus, das Sie verwalten möchten. Die Rollen für Ihren Benutzer werden angezeigt. Wählen Sie

Zugriffstasten, um Anmeldeinformationen für den Befehlszeilen- oder programmgesteuerten Zugriff für den entsprechenden Berechtigungssatz zu erhalten. Verwenden Sie den vordefinierten `PowerUserAccess` Berechtigungssatz oder den von Ihnen oder Ihrem Arbeitgeber erstellten Berechtigungssatz, um Berechtigungen mit den geringsten Rechten für die Entwicklung anzuwenden.

b. Wählen Sie im Dialogfeld Anmeldeinformationen abrufen entweder MacOS und Linux oder Windows aus (je nach dem Betriebssystem.

c. Wählen Sie die IAM Identity Center-Anmeldedatenmethode, um die Daten `Issuer URL` und `SSO Region` Werte zu erhalten, die Sie für den nächsten Schritt benötigen. Hinweis: `SSO Start URL` kann synonym mit verwendet werden. `Issuer URL`

3. Führen AWS CLI Sie den Befehl in der Befehlszeile aus. `aws configure sso` Wenn Sie dazu aufgefordert werden, geben Sie die Konfigurationswerte ein, die Sie im vorherigen Schritt gesammelt haben. Einzelheiten zu diesem AWS CLI Befehl finden Sie unter Konfigurieren Ihres Profils mit dem `aws configure sso` Assistenten.

   a. Geben Sie für die Aufforderung den Wert ein`SSO Start URL`, den Sie für erhalten haben`Issuer URL`.

   b. Wir empfehlen, den CLI-Profilnamen einzugeben, *`default`* wenn Sie beginnen. Informationen darüber, wie Sie nicht standardmäßige (benannte) Profile und die zugehörige Umgebungsvariable einrichten, finden Sie unterProfile.

4. (Optional) Bestätigen Sie in der AWS CLI Befehlszeile die Identität der aktiven Sitzung, indem Sie den `aws sts get-caller-identity` Befehl ausführen. In der Antwort sollte der von Ihnen konfigurierte IAM Identity Center-Berechtigungssatz angezeigt werden.

5. Wenn Sie ein AWS SDK verwenden, erstellen Sie eine Anwendung für Ihr SDK in Ihrer Entwicklungsumgebung.

   a. In einigen SDKs Fällen `SSOOIDC` müssen zusätzliche Pakete wie `SSO` und zu Ihrer Anwendung hinzugefügt werden, bevor Sie die IAM Identity Center-Authentifizierung verwenden können. Einzelheiten finden Sie in Ihrem spezifischen SDK.

   b. Wenn Sie den Zugriff auf zuvor konfiguriert haben AWS, überprüfen Sie Ihre geteilte AWS `credentials` Datei auf etwaigeAWS Zugriffstasten. Aufgrund der Machen Sie sich mit der Kette der Anbieter von Anmeldeinformationen vertraut Rangfolge müssen Sie alle statischen Anmeldeinformationen entfernen, bevor das SDK oder das Tool die IAM Identity Center-Anmeldeinformationen verwendet.

Einen ausführlichen Einblick in die Verwendung SDKs und Aktualisierung der Anmeldeinformationen mithilfe dieser Konfiguration durch die Tools finden Sie unter. Wie die IAM Identity Center-Authentifizierung gelöst wird AWS SDKs und welche Tools

Informationen zur Konfiguration der IAM Identity Center-Provider-Einstellungen direkt in der gemeinsam genutzten `config` Datei finden Sie IAM Identity Center-Anmeldeinformationsanbieter in diesem Handbuch.

Aktualisierung der Portalzugriffssitzungen

Ihr Zugriff läuft irgendwann ab und beim SDK oder Tool tritt ein Authentifizierungsfehler auf. Wann dieser Ablauf eintritt, hängt von Ihrer konfigurierten Sitzungsdauer ab. Um die Access-Portal-Sitzung bei Bedarf erneut zu aktualisieren, verwenden Sie den, AWS CLI um den `aws sso login` Befehl auszuführen.

Sie können sowohl die Sitzungsdauer des IAM Identity Center-Zugriffsportals als auch die Sitzungsdauer des Berechtigungssatzes verlängern. Dadurch verlängert sich die Zeit, in der Sie Code ausführen können, bevor Sie sich erneut manuell mit dem anmelden müssen. AWS CLI Weitere Informationen finden Sie in folgenden Themen im AWS IAM Identity Center -Benutzerhandbuch:

- Dauer der IAM Identity Center-Sitzung — Konfigurieren Sie die Dauer der AWS Zugriffsportalsitzungen Ihrer Benutzer
- Sitzungsdauer mit Zugriffsrechten — Legen Sie die Sitzungsdauer fest

Wie die IAM Identity Center-Authentifizierung gelöst wird AWS SDKs und welche Tools

Relevante IAM Identity Center-Bedingungen

Die folgenden Begriffe helfen Ihnen, den Prozess und die Konfiguration dahinter AWS IAM Identity Center zu verstehen. In der Dokumentation für AWS das SDK werden für einige dieser Authentifizierungskonzepte andere Namen als für IAM Identity Center APIs verwendet. Es ist hilfreich, beide Namen zu kennen.

Die folgende Tabelle zeigt, in welcher Beziehung alternative Namen zueinander stehen.

Name des IAM Identity Center	SDK-API-Name	Description
Identity Center	`sso`	Obwohl AWS Single Sign-On umbenannt wurde, behalten die `sso` API-Namespaces aus Gründen der Abwärtskompatibilität ihren ursprünglichen Namen. Weitere Informationen finden Sie unter Umbenennung von IAM Identity Center im Benutzerhandbuch zu AWS IAM Identity Center .
IAM Identity Center-Konsole Administrationskonsole		Die Konsole, mit der Sie Single Sign-On konfigurieren.
AWS auf die Portal-URL zugreifen		Eine eindeutige URL für Ihr IAM Identity Center-Konto, wie`https://xxx.awsapps.com/start` . Sie melden sich mit Ihren IAM Identity Center-Anmeldeinformationen bei diesem Portal an.
Sitzung des IAM Identity Center-Zugriffsportals	Authentifizierungssitzung	Stellt dem Anrufer ein Bearer-Zugriffstoken zur Verfügung.
Sitzung mit Berechtigungssatz		Die IAM-Sitzung, die das SDK intern für die AWS-Service Aufrufe verwendet. In informellen Diskussionen wird dies möglicherweise fälschlicherweise als „Rollensitzung“ bezeichnet.

Name des IAM Identity Center	SDK-API-Name	Description
Anmeldeinformationen für den Berechtigungssatz	AWS Anmeldeinformationen Sigv4-Anmeldeinformationen	Die Anmeldeinformationen, die das SDK tatsächlich für die meisten AWS-Service Aufrufe verwendet (insbesondere für alle AWS-Service Sigv4-Aufrufe). In informellen Diskussionen werden Sie möglicherweise feststellen, dass dies fälschlicherweise als „Rollenanmeldedaten“ bezeichnet wird.
Anbieter von IAM Identity Center-Anmeldeinformationen	Anbieter von SSO-Anmeldeinformationen	Wie Sie die Anmeldeinformationen erhalten, z. B. die Klasse oder das Modul, das die Funktionalität bereitstellt.

Erfahren Sie mehr über die Auflösung von SDK-Anmeldeinformationen für AWS-Services

Die IAM Identity Center-API tauscht Inhaber-Token-Anmeldeinformationen gegen Sigv4-Anmeldeinformationen aus. Die meisten AWS-Services sind Sigv4 APIs, mit einigen Ausnahmen wie und. Amazon CodeWhisperer Amazon CodeCatalyst Im Folgenden wird der Prozess zur Auflösung von Anmeldeinformationen beschrieben, mit dem die meisten AWS-Service Aufrufe für Ihren Anwendungscode unterstützt werden. AWS IAM Identity Center

Starten Sie eine AWS Access-Portal-Sitzung

- Starten Sie den Vorgang, indem Sie sich mit Ihren Anmeldeinformationen bei der Sitzung anmelden.
  - Verwenden Sie den `aws sso login` Befehl in der AWS Command Line Interface (AWS CLI). Dadurch wird eine neue IAM Identity Center-Sitzung gestartet, falls Sie noch keine aktive Sitzung haben.
- Wenn Sie eine neue Sitzung starten, erhalten Sie vom IAM Identity Center ein Aktualisierungs- und Zugriffstoken. AWS CLI Außerdem wird eine SSO-Cache-JSON-Datei mit einem neuen

Zugriffstoken und einem Aktualisierungstoken aktualisiert und für die Verwendung durch SDKs verfügbar gemacht.

- Wenn Sie bereits eine aktive Sitzung haben, verwendet der AWS CLI Befehl die bestehende Sitzung erneut und läuft ab, sobald die bestehende Sitzung abläuft. Informationen zum Einstellen der Länge einer IAM Identity Center-Sitzung finden Sie im Benutzerhandbuch unter Konfigurieren der Dauer der AWS Access-Portal-Sitzungen Ihrer AWS IAM Identity Center Benutzer.
  - Die maximale Sitzungsdauer wurde auf 90 Tage verlängert, um die Notwendigkeit häufiger Anmeldungen zu reduzieren.

Wie erhält das SDK Anmeldeinformationen für Anrufe AWS-Service

SDKs bietet Zugriff auf, AWS-Services wenn Sie ein Client-Objekt pro Dienst instanziieren. Wenn das ausgewählte Profil der gemeinsam genutzten AWS `config` Datei für die Auflösung von IAM Identity Center-Anmeldeinformationen konfiguriert ist, wird IAM Identity Center zur Auflösung der Anmeldeinformationen für Ihre Anwendung verwendet.

- Der Prozess zur Auflösung der Anmeldeinformationen wird während der Laufzeit abgeschlossen, wenn ein Client erstellt wird.

Um Anmeldeinformationen für Sigv4 APIs mithilfe von IAM Identity Center Single Sign-On abzurufen, verwendet das SDK das IAM Identity Center-Zugriffstoken, um eine IAM-Sitzung aufzurufen. Diese IAM-Sitzung wird als Berechtigungssatz-Sitzung bezeichnet und ermöglicht den AWS Zugriff auf das SDK, indem sie eine IAM-Rolle übernimmt.

- Die Sitzungsdauer des Berechtigungssatzes wird unabhängig von der Dauer der IAM Identity Center-Sitzung festgelegt.
  - Informationen zum Einstellen der Sitzungsdauer mit dem Berechtigungssatz finden Sie unter Sitzungsdauer festlegen im AWS IAM Identity Center Benutzerhandbuch.
- Beachten Sie, dass die Berechtigungssatz-Anmeldeinformationen in den meisten AWS SDK-API-Dokumentationen auch als AWS Anmeldeinformationen und Sigv4-Anmeldeinformationen bezeichnet werden.

Die Anmeldeinformationen für den Berechtigungssatz werden bei einem Aufruf getRoleCredentialsder IAM Identity Center-API an das SDK zurückgegeben. Das Client-Objekt des SDK verwendet diese angenommene IAM-Rolle, um Aufrufe an das zu tätigen AWS-Service, z. B. Amazon

S3 aufzufordern, die Buckets in Ihrem Konto aufzulisten. Das Client-Objekt kann mit diesen Berechtigungssatz-Anmeldeinformationen weiterarbeiten, bis die Berechtigungssatz-Sitzung abläuft.

Ablauf und Aktualisierung der Sitzung

Bei Verwendung von wird das Konfiguration des SSO-Token-Anbieters vom IAM Identity Center abgerufene stündliche Zugriffstoken automatisch mit dem Aktualisierungstoken aktualisiert.

- Wenn das Zugriffstoken abgelaufen ist, wenn das SDK versucht, es zu verwenden, verwendet das SDK das Aktualisierungstoken, um zu versuchen, ein neues Zugriffstoken abzurufen. Das IAM Identity Center vergleicht das Aktualisierungstoken mit der Sitzungsdauer Ihres IAM Identity Center-Zugriffsportals. Wenn das Aktualisierungstoken nicht abgelaufen ist, antwortet das IAM Identity Center mit einem anderen Zugriffstoken.
- Dieses Zugriffstoken kann entweder verwendet werden, um die Berechtigungssatz-Sitzung vorhandener Clients zu aktualisieren oder um Anmeldeinformationen für neue Clients aufzulösen.

Wenn die Sitzung des IAM Identity Center-Zugriffsportals jedoch abgelaufen ist, wird kein neues Zugriffstoken gewährt. Daher kann die Dauer des Berechtigungssatzes nicht verlängert werden. Sie läuft ab (und der Zugriff geht verloren), wenn die Dauer der zwischengespeicherten Berechtigungssatz-Sitzung für bestehende Clients überschritten wird.

Bei jedem Code, der einen neuen Client erstellt, schlägt die Authentifizierung fehl, sobald die IAM Identity Center-Sitzung abläuft. Das liegt daran, dass die Anmeldeinformationen für den Berechtigungssatz nicht zwischengespeichert werden. Ihr Code kann erst dann einen neuen Client erstellen und die Auflösung der Anmeldeinformationen abschließen, wenn Sie über ein gültiges Zugriffstoken verfügen.

Um es noch einmal zusammenzufassen: Wenn das SDK neue Berechtigungssatz-Anmeldeinformationen benötigt, sucht das SDK zunächst nach gültigen, vorhandenen Anmeldeinformationen und verwendet diese. Dies gilt unabhängig davon, ob die Anmeldeinformationen für einen neuen Client oder für einen vorhandenen Client mit abgelaufenen Anmeldeinformationen bestimmt sind. Wenn Anmeldeinformationen nicht gefunden werden oder sie nicht gültig sind, ruft das SDK die IAM Identity Center-API auf, um neue Anmeldeinformationen abzurufen. Um die API aufzurufen, benötigt sie das Zugriffstoken. Wenn das Zugriffstoken abgelaufen ist, verwendet das SDK das Aktualisierungstoken, um ein neues Zugriffstoken vom IAM Identity Center-Dienst abzurufen. Dieses Token wird gewährt, wenn Ihre IAM Identity Center-Zugriffsportalsitzung nicht abgelaufen ist.

Authentifizierung AWS SDKs und Tools mit IAM Roles Anywhere

Sie können IAM Roles Anywhere verwenden, um temporäre Sicherheitsanmeldeinformationen in IAM für Workloads wie Server, Container und Anwendungen abzurufen, die außerhalb von ausgeführt werden. AWS Um IAM Roles Anywhere verwenden zu können, müssen Ihre Workloads X.509-Zertifikate verwenden. Ihr Cloud-Administrator sollte das Zertifikat und den privaten Schlüssel bereitstellen, die für die Konfiguration von IAM Roles Anywhere als Ihren Anmeldeinformationsanbieter erforderlich sind.

Schritt 1: IAM-Rollen Anywhere konfigurieren

IAM Roles Anywhere bietet eine Möglichkeit, temporäre Anmeldeinformationen für einen Workload oder Prozess abzurufen, der außerhalb von ausgeführt wird. AWS Bei der Zertifizierungsstelle wird ein Vertrauensanker eingerichtet, um temporäre Anmeldeinformationen für die zugehörige IAM-Rolle abzurufen. Die Rolle legt die Berechtigungen fest, über die Ihr Workload verfügt, wenn Ihr Code bei IAM Roles Anywhere authentifiziert wird.

Schritte zum Einrichten des Vertrauensankers, der IAM-Rolle und des IAM Roles Anywhere-Profils finden Sie unter Einen Vertrauensanker und ein Profil in AWS Identity and Access Management Roles Anywhere erstellen im IAM Roles Anywhere-Benutzerhandbuch.

> **Note**
>
> Ein Profil im IAM Roles Anywhere-Benutzerhandbuch bezieht sich auf ein einzigartiges Konzept innerhalb des IAM Roles Anywhere-Dienstes. Es hat nichts mit den Profilen in der gemeinsam genutzten AWS `config` Datei zu tun.

Schritt 2: Verwenden Sie IAM Roles Anywhere

Verwenden Sie das Credential Helper-Tool von IAM Roles Anywhere, um temporäre Sicherheitsanmeldedaten von IAM Roles Anywhere abzurufen. Das Credential Tool implementiert den Signaturprozess für IAM Roles Anywhere.

Anweisungen zum Herunterladen des Credential Helpertools finden Sie unter Abrufen temporärer Sicherheitsanmeldedaten von AWS Identity and Access Management Roles Anywhere im IAM Roles Anywhere-Benutzerhandbuch.

Um temporäre Sicherheitsanmeldedaten von IAM Roles Anywhere mit AWS SDKs und dem zu verwenden AWS CLI, können Sie die `credential_process` Einstellung in der gemeinsam genutzten Datei konfigurieren. AWS `config` Der SDKs und AWS CLI unterstützt einen Prozessanmeldedienstanbieter, der `credential_process` zur Authentifizierung verwendet wird. Im Folgenden wird die allgemeine Struktur dargestellt, die festgelegt werden muss. `credential_process`

```
credential_process = [path to helper tool] [command] [--parameter1 value] [--
parameter2 value] [...]
```

Der `credential-process` Befehl des Hilfstools gibt temporäre Anmeldeinformationen in einem Standard-JSON-Format zurück, das mit der `credential_process` Einstellung kompatibel ist. Beachten Sie, dass der Befehlsname einen Bindestrich enthält, der Einstellungsname jedoch einen Unterstrich. Der Befehl erfordert die folgenden Parameter:

- `private-key`— Der Pfad zu dem privaten Schlüssel, der die Anfrage signiert hat.
- `certificate`— Der Pfad zum Zertifikat.
- `role-arn`— Der ARN der Rolle, für die temporäre Anmeldeinformationen abgerufen werden sollen.
- `profile-arn`— Der ARN des Profils, das eine Zuordnung für die angegebene Rolle bereitstellt.
- `trust-anchor-arn`— Der ARN des Vertrauensankers, der zur Authentifizierung verwendet wurde.

Ihr Cloud-Administrator sollte das Zertifikat und den privaten Schlüssel bereitstellen. Alle drei ARN-Werte können aus dem kopiert werden AWS-Managementkonsole. Das folgende Beispiel zeigt eine gemeinsam genutzte `config` Datei, in der das Abrufen temporärer Anmeldeinformationen aus dem Hilfstool konfiguriert wird.

```
[profile dev]
credential_process = ./aws_signing_helper credential-process --certificate /
path/to/certificate --private-key /path/to/private-key --trust-anchor-
arn arn:aws:rolesanywhere:region:account:trust-anchor/TA_ID --profile-
arn arn:aws:rolesanywhere:region:account:profile/PROFILE_ID --role-
arn arn:aws:iam::account:role/ROLE_ID
```

Optionale Parameter und weitere Informationen zum Hilfstool finden Sie unter IAM Roles Anywhere Credential Helper on. GitHub

Einzelheiten zur SDK-Konfigurationseinstellung selbst und zum Anbieter von Prozessanmeldedaten finden Sie Anbieter von Prozessanmeldeinformationen in diesem Handbuch.

Übernahme einer Rolle mit AWS Anmeldeinformationen zur Authentifizierung AWS SDKs und Tools

Die Übernahme einer Rolle beinhaltet die Verwendung einer Reihe temporärer Sicherheitsanmeldedaten für den Zugriff auf AWS Ressourcen, auf die Sie sonst möglicherweise keinen Zugriff hätten. Diese temporären Anmeldeinformationen bestehen aus einer Zugriffsschlüssel-ID, einem geheimen Zugriffsschlüssel und einem Sicherheits-Token. Weitere Informationen zu AWS -Security-Token-Service (AWS STS) API-Anfragen finden Sie in der AWS -Security-Token-Service API-Referenz unter Aktionen.

Um Ihr SDK oder Tool so einzurichten, dass es eine Rolle übernimmt, müssen Sie zunächst eine bestimmte Rolle erstellen oder identifizieren, die Sie übernehmen möchten. IAM-Rollen werden durch eine Rolle mit dem Amazon Resource Name (ARN) eindeutig identifiziert. Rollen bauen Vertrauensbeziehungen zu einer anderen Entität auf. Bei der vertrauenswürdigen Entität, die die Rolle verwendet, kann es sich um die eine AWS-Service oder andere handeln AWS-Konto. Weitere Informationen zu IAM-Rollen finden Sie unter Verwenden von IAM-Rollen im IAM-Benutzerhandbuch.

Nachdem die IAM-Rolle identifiziert wurde und Sie diese Rolle als vertrauenswürdig eingestuft haben, können Sie Ihr SDK oder Tool so konfigurieren, dass die von der Rolle gewährten Berechtigungen verwendet werden.

> **Note**
>
> Es hat sich AWS bewährt, wann immer möglich regionale Endpunkte zu verwenden und Ihre zu konfigurieren. AWS-Region

Nehmen Sie eine IAM-Rolle an

AWS STS Gibt bei Übernahme einer Rolle einen Satz temporärer Sicherheitsanmeldedaten zurück. Diese Anmeldeinformationen stammen aus einem anderen Profil oder aus der Instance oder dem Container, in dem Ihr Code ausgeführt wird. Am häufigsten wird diese Art der Rollenübernahme verwendet, wenn Sie über AWS Anmeldeinformationen für ein Konto verfügen, Ihre Anwendung jedoch Zugriff auf Ressourcen in einem anderen Konto benötigt.

Schritt 1: Richten Sie eine IAM-Rolle ein

Um Ihr SDK oder Tool so einzurichten, dass es eine Rolle übernimmt, müssen Sie zunächst eine bestimmte Rolle erstellen oder identifizieren, die Sie übernehmen möchten. IAM-Rollen werden mithilfe eines Rollen-ARN eindeutig identifiziert. Rollen stellen Vertrauensbeziehungen zu einer anderen Entität her, in der Regel innerhalb Ihres Kontos oder für kontoübergreifenden Zugriff. Informationen zur Einrichtung finden Sie unter IAM-Rollen erstellen im IAM-Benutzerhandbuch.

Schritt 2: Konfigurieren Sie das SDK oder das Tool

Konfigurieren Sie das SDK oder das Tool so, dass Anmeldeinformationen von `credential_source` oder abgerufen `source_profile` werden.

Wird verwendet`credential_source`, um Anmeldeinformationen aus einem Amazon ECS-Container, einer Amazon EC2 EC2-Instance oder aus Umgebungsvariablen zu beziehen.

Wird verwendet`source_profile`, um Anmeldeinformationen aus einem anderen Profil zu beziehen. `source_profile`unterstützt auch Rollenverkettung, d. h. Hierarchien von Profilen, bei denen eine übernommene Rolle dann verwendet wird, um eine andere Rolle anzunehmen.

Wenn Sie dies in einem Profil angeben, führt das SDK oder Tool automatisch den entsprechenden AWS STS `AssumeRole`API-Aufruf für Sie durch. Um temporäre Anmeldeinformationen abzurufen und zu verwenden, indem Sie eine Rolle übernehmen, geben Sie die folgenden Konfigurationswerte in der gemeinsam genutzten AWS `config` Datei an. Weitere Informationen zu den einzelnen Einstellungen finden Sie im Nehmen Sie die Einstellungen des Anbieters für Anmeldeinformationen an Abschnitt.

- `role_arn`— Aus der IAM-Rolle, die Sie in Schritt 1 erstellt haben
- Konfigurieren Sie entweder oder `credential_source` `source_profile`
- (Optional) `duration_seconds`
- (Optional) `external_id`
- (Optional) `mfa_serial`
- (Optional) `role_session_name`

Die folgenden Beispiele zeigen die Konfiguration der beiden Optionen zur Übernahme von Rollen in einer gemeinsam genutzten `config` Datei:

```
role_arn = arn:aws:iam::123456789012:role/my-role-name
```

```
credential_source = Ec2InstanceMetadata
```

```
[profile-with-user-that-can-assume-role]
aws_access_key_id=AKIAIOSFODNN7EXAMPLE
aws_secret_access_key=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
aws_session_token=IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZVERYLONGSTRI

[profile dev]
region = us-east-1
output = json
role_arn = arn:aws:iam::123456789012:role/my-role-name
source_profile = profile-with-user-that-can-assume-role
role_session_name = my_session
```

Einzelheiten zu allen Einstellungen des Anbieters für die Übernahme der Rollenanmeldedaten finden Sie Übernehmen Sie die Rolle Credential Provider in diesem Handbuch.

Übernahme einer Rolle bei Web Identity oder OpenID Connect zur Authentifizierung und Tools AWS SDKs

Die Übernahme einer Rolle beinhaltet die Verwendung einer Reihe temporärer Sicherheitsanmeldedaten für den Zugriff auf AWS Ressourcen, auf die Sie sonst möglicherweise keinen Zugriff hätten. Diese temporären Anmeldeinformationen bestehen aus einer Zugriffsschlüssel-ID, einem geheimen Zugriffsschlüssel und einem Sicherheits-Token. Weitere Informationen zu AWS -Security-Token-Service (AWS STS) API-Anfragen finden Sie in der AWS -Security-Token-Service API-Referenz unter Aktionen.

Um Ihr SDK oder Tool so einzurichten, dass es eine Rolle übernimmt, müssen Sie zunächst eine bestimmte Rolle erstellen oder identifizieren, die Sie übernehmen möchten. IAM-Rollen werden durch eine Rolle mit dem Amazon Resource Name (ARN) eindeutig identifiziert. Rollen bauen Vertrauensbeziehungen zu einer anderen Entität auf. Bei der vertrauenswürdigen Entität, die die Rolle verwendet, kann es sich um einen Web-Identitätsanbieter, OpenID Connect (OIDC) oder einen SAML-Verbund handeln. Weitere Informationen zu IAM-Rollen finden Sie unter Methoden zur Übernahme einer Rolle im IAM-Benutzerhandbuch.

Nachdem die IAM-Rolle in Ihrem SDK konfiguriert wurde und diese Rolle so konfiguriert ist, dass sie Ihrem Identitätsanbieter vertraut, können Sie Ihr SDK weiter so konfigurieren, dass es diese Rolle übernimmt, um temporäre Anmeldeinformationen zu erhalten. AWS

> **Note**
>
> Es hat sich AWS bewährt, wann immer möglich regionale Endpunkte zu verwenden und Ihre zu konfigurieren. AWS-Region

Verbunden mit Web-Identität oder OpenID Connect

Sie können die JSON-Web-Tokens (JWTs) von öffentlichen Identitätsanbietern wie Login With Amazon, Facebook, Google verwenden, um temporäre AWS Anmeldeinformationen mithilfe von`AssumeRoleWithWebIdentity`. Je nachdem, wie sie verwendet werden, JWTs können diese als ID-Token oder Zugriffstoken bezeichnet werden. Sie können auch von Identitätsanbietern (IdPs) JWTs ausgegebene Daten verwenden, die mit dem Discovery-Protokoll von OIDC kompatibel sind, z. B. EntraId oder. PingFederate

Wenn Sie Amazon Elastic Kubernetes Service verwenden, bietet diese Funktion die Möglichkeit, unterschiedliche IAM-Rollen für jedes Ihrer Dienstkonten in einem Amazon EKS-Cluster anzugeben. Diese Kubernetes-Funktion verteilt sie an Ihre Pods JWTs , die dann von diesem Anmeldeinformationsanbieter verwendet werden, um temporäre Anmeldeinformationen abzurufen. AWS Weitere Informationen zu dieser Amazon EKS-Konfiguration finden Sie unter IAM-Rollen für Dienstkonten im Amazon EKS-Benutzerhandbuch. Für eine einfachere Option empfehlen wir jedoch, stattdessen Amazon EKS Pod Identities zu verwenden, sofern Ihr SDK dies unterstützt.

Schritt 1: Richten Sie einen Identitätsanbieter und eine IAM-Rolle ein

Um den Verbund mit einem externen IdP zu konfigurieren, verwenden Sie einen IAM-Identitätsanbieter, um AWS Informationen über den externen IdP und seine Konfiguration zu erhalten. Dadurch wird Vertrauen zwischen Ihrem AWS-Konto und dem externen IdP hergestellt. Bevor Sie das SDK für die Verwendung des JSON Web Tokens (JWT) für die Authentifizierung konfigurieren, müssen Sie zunächst den Identitätsanbieter (IdP) und die IAM-Rolle einrichten, die für den Zugriff verwendet wird. Informationen zur Einrichtung finden Sie unter Erstellen einer Rolle für Web-Identität oder OpenID Connect Federation (Konsole) im IAM-Benutzerhandbuch.

Schritt 2: Konfigurieren Sie das SDK oder das Tool

Konfigurieren Sie das SDK oder das Tool so, dass es ein JSON Web Token (JWT) von AWS STS für die Authentifizierung verwendet.

Wenn Sie dies in einem Profil angeben, führt das SDK oder Tool automatisch den entsprechenden AWS STS `AssumeRoleWithWebIdentity`API-Aufruf für Sie durch. Um temporäre Anmeldeinformationen mithilfe des Web Identity Federation abzurufen und zu verwenden, geben Sie die folgenden Konfigurationswerte in der gemeinsam genutzten AWS `config` Datei an. Weitere Informationen zu den einzelnen Einstellungen finden Sie im Nehmen Sie die Einstellungen des Anbieters für Anmeldeinformationen an Abschnitt.

- `role_arn`— Aus der IAM-Rolle, die Sie in Schritt 1 erstellt haben
- `web_identity_token_file`- Vom externen IdP
- (Optional) `duration_seconds`
- (Optional) `role_session_name`

Im Folgenden finden Sie ein Beispiel für eine Konfiguration einer gemeinsam genutzten `config` Datei, bei der eine Rolle mit Web-Identität übernommen wird:

```
[profile web-identity]
role_arn=arn:aws:iam::123456789012:role/my-role-name
web_identity_token_file=/path/to/a/token
```

> **Note**
>
> Erwägen Sie für mobile Anwendungen die Verwendung von Amazon Cognito. Amazon Cognito fungiert als Identitätsbroker und erledigt einen Großteil der Verbundarbeit für Sie. Der Amazon Cognito Cognito-Identitätsanbieter ist jedoch nicht wie andere Identitätsanbieter in den Kernbibliotheken SDKs und Tools enthalten. Um auf die Amazon Cognito-API zuzugreifen, schließen Sie den Amazon Cognito-Serviceclient in den Build oder die Bibliotheken für Ihr SDK oder Tool ein. Informationen zur Verwendung mit AWS SDKs finden Sie unter Codebeispiele im Amazon Cognito Developer Guide.

Einzelheiten zu allen Einstellungen des Anbieters von Anmeldedaten für die Übernahme einer Rolle finden Sie Übernehmen Sie die Rolle Credential Provider in diesem Handbuch.

Verwendung von AWS Zugriffsschlüsseln zur Authentifizierung AWS SDKs und Tools

Die Verwendung von AWS Zugriffsschlüsseln ist eine Option für die Authentifizierung bei der Verwendung von AWS SDKs Tools.

Verwenden Sie kurzfristige Anmeldeinformationen

Wir empfehlen, Ihr SDK oder Tool so zu konfigurieren, Verwenden von IAM Identity Center zur Authentifizierung von AWS SDK und Tools dass Optionen für die erweiterte Sitzungsdauer verwendet werden.

Informationen zum direkten Einrichten der temporären Anmeldeinformationen des SDK oder Tools finden Sie jedoch unterVerwendung von kurzfristigen Anmeldeinformationen zur Authentifizierung AWS SDKs und Tools.

Verwenden Sie langfristige Anmeldeinformationen

> **Warning**
>
> Um Sicherheitsrisiken zu vermeiden, sollten Sie IAM-Benutzer nicht zur Authentifizierung verwenden, wenn Sie speziell entwickelte Software entwickeln oder mit echten Daten arbeiten. Verwenden Sie stattdessen den Verbund mit einem Identitätsanbieter wie AWS IAM Identity Center.

Verwalten Sie den Zugriff auf alle AWS-Konten

Aus Sicherheitsgründen empfehlen wir die Verwendung AWS Organizations zusammen mit IAM Identity Center, um den Zugriff für alle Ihre AWS-Konten Benutzer zu verwalten. Weitere Informationen finden Sie unter Bewährte Methoden für die Sicherheit in IAM im IAM-Benutzerhandbuch.

Sie können Benutzer im IAM Identity Center erstellen, Microsoft Active Directory verwenden, einen SAML 2.0-Identitätsanbieter (IdP) verwenden oder Ihren IdP individuell mit einem Verbund verbinden. AWS-Konten Mit einem dieser Ansätze können Sie Ihren Benutzern ein Single-Sign-On-Erlebnis bieten. Sie können auch die Multi-Faktor-Authentifizierung (MFA) erzwingen und temporäre

Anmeldeinformationen für AWS-Konto den Zugriff verwenden. Dies unterscheidet sich von einem IAM-Benutzer, bei dem es sich um langfristige Anmeldeinformationen handelt, die gemeinsam genutzt werden können und die das Sicherheitsrisiko für Ihre Ressourcen erhöhen können. AWS

Erstellen Sie IAM-Benutzer nur für Sandbox-Umgebungen

Wenn Sie noch keine Erfahrung damit haben AWS, können Sie einen IAM-Testbenutzer erstellen und ihn dann verwenden, um Tutorials durchzuführen und zu erkunden, was zu bieten AWS ist. Es ist in Ordnung, diese Art von Anmeldeinformationen zu verwenden, wenn Sie lernen, aber wir empfehlen, sie nicht außerhalb einer Sandbox-Umgebung zu verwenden.

Für die folgenden Anwendungsfälle kann es sinnvoll sein, mit IAM-Benutzern zu beginnen in: AWS

- Erste Schritte mit Ihrem AWS SDK oder Tool und Erkundung AWS-Services in einer Sandbox-Umgebung
- Das Ausführen von geplanten Skripten, Jobs und anderen automatisierten Prozessen, die keinen manuellen Anmeldevorgang unterstützen, ist Teil Ihres Lernprozesses.

Wenn Sie IAM-Benutzer außerhalb dieser Anwendungsfälle verwenden, wechseln Sie so schnell wie möglich zu IAM Identity Center oder binden Sie Ihren Identitätsanbieter an. AWS-Konten Weitere Informationen finden Sie unter Identitätsverbund in. AWS

Sichere IAM-Benutzerzugriffsschlüssel

Sie sollten die IAM-Benutzerzugriffsschlüssel regelmäßig wechseln. Folgen Sie den Anweisungen unter Rotierende Zugriffsschlüssel im IAM-Benutzerhandbuch. Wenn Sie glauben, dass Sie versehentlich Ihre IAM-Benutzerzugriffsschlüssel geteilt haben, wechseln Sie Ihre Zugriffsschlüssel.

Die IAM-Benutzerzugriffsschlüssel sollten in der gemeinsam genutzten AWS `credentials` Datei auf dem lokalen Computer gespeichert werden. Speichern Sie die IAM-Benutzerzugriffsschlüssel nicht in Ihrem Code. Fügen Sie keiner Quellcodeverwaltungssoftware Konfigurationsdateien hinzu, die Ihre IAM-Benutzerzugriffsschlüssel enthalten. Externe Tools, wie das Open-Source-Projekt git-secrets, können Sie davor schützen, versehentlich vertrauliche Informationen in ein Git-Repository zu übertragen. Weitere Informationen finden Sie unter IAM-Identitäten (Benutzer, Benutzergruppen und Rollen) im -IAM-Benutzerhandbuch.

Informationen zum Einrichten eines IAM-Benutzers für die ersten Schritte finden Sie unter. Verwendung langfristiger Anmeldeinformationen zur Authentifizierung AWS SDKs und Tools

Verwendung von kurzfristigen Anmeldeinformationen zur Authentifizierung AWS SDKs und Tools

Wir empfehlen, Ihr AWS SDK oder Tool so zu konfigurieren, dass es Verwenden von IAM Identity Center zur Authentifizierung von AWS SDK und Tools mit Optionen für die erweiterte Sitzungsdauer verwendet wird. Sie können jedoch temporäre Anmeldeinformationen, die im AWS Access Portal verfügbar sind, kopieren und verwenden. Wenn diese Anmeldeinformationen ablaufen, müssen neue kopiert werden. Sie können die temporären Anmeldeinformationen in einem Profil verwenden oder sie als Werte für Systemeigenschaften und Umgebungsvariablen verwenden.

Bewährte Methode: Anstatt die Zugriffsschlüssel und ein Token in der Anmeldeinformationsdatei manuell zu verwalten, empfehlen wir, dass Ihre Anwendung temporäre Anmeldeinformationen verwendet, die bereitgestellt werden von:

- Ein AWS Rechenservice, z. B. das Ausführen Ihrer Anwendung auf Amazon Elastic Compute Cloud oder in AWS Lambda.
- Eine weitere Option in der Kette der Anmeldeinformationsanbieter, wie Verwenden von IAM Identity Center zur Authentifizierung von AWS SDK und Tools z.
- Oder verwenden Sie dieAnbieter von Prozessanmeldeinformationen, um temporäre Anmeldeinformationen abzurufen.

Richten Sie eine Anmeldeinformationsdatei mit kurzfristigen Anmeldeinformationen ein, die Sie aus dem AWS Access Portal abgerufen haben

1. Erstellen Sie eine Datei mit gemeinsamen Anmeldeinformationen.
2. Fügen Sie in der Anmeldeinformationsdatei den folgenden Platzhaltertext ein, bis Sie funktionierende temporäre Anmeldeinformationen einfügen.

```
[default]
aws_access_key_id=<value from AWS access portal>
aws_secret_access_key=<value from AWS access portal>
aws_session_token=<value from AWS access portal>
```

3. Speichern Sie die Datei. Die Datei `~/.aws/credentials` sollte jetzt auf Ihrem lokalen Entwicklungssystem vorhanden sein. Diese Datei enthält das [Standard-] Profil, das das SDK oder das Tool verwendet, wenn kein bestimmtes benanntes Profil angegeben ist.
4. Melden Sie sich beim AWS Access-Portal an.

5. Folgen Sie diesen Anweisungen zur manuellen Aktualisierung der Anmeldeinformationen, um die Anmeldeinformationen der IAM-Rolle aus dem AWS Zugriffsportal zu kopieren.

   a. Wählen Sie für Schritt 4 der verlinkten Anleitung den IAM-Rollennamen aus, der den Zugriff für Ihre Entwicklungsanforderungen gewährt. Diese Rolle hat normalerweise einen Namen wie PowerUserAccessoder Developer.

   b. Wählen Sie für Schritt 7 in den verlinkten Anweisungen die Option Manuelles Hinzufügen eines Profils zu Ihrer AWS Anmeldeinformationsdatei aus und kopieren Sie den Inhalt.

6. Fügen Sie die kopierten Anmeldeinformationen in Ihre lokale `credentials` Datei ein. Der generierte Profilname wird nicht benötigt, wenn Sie das `default` Profil verwenden. Ihre Datei sollte wie folgt aussehen.

```
[default]
aws_access_key_id=AKIAIOSFODNN7EXAMPLE
aws_secret_access_key=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
aws_session_token=IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZVERYLONG
```

7. Speichern Sie die `credentials`-Datei.

Wenn das SDK einen Service-Client erstellt, greift es auf diese temporären Anmeldeinformationen zu und verwendet sie für jede Anfrage. Die in Schritt 5a ausgewählten Einstellungen für die IAM-Rolle bestimmen, wie lange die temporären Anmeldeinformationen gültig sind. Die maximale Dauer beträgt zwölf Stunden.

Wenn die temporären Anmeldeinformationen abgelaufen sind, wiederholen Sie die Schritte 4 bis 7.

Verwendung langfristiger Anmeldeinformationen zur Authentifizierung AWS SDKs und Tools

> **Warning**
>
> Um Sicherheitsrisiken zu vermeiden, sollten Sie IAM-Benutzer nicht zur Authentifizierung verwenden, wenn Sie speziell entwickelte Software entwickeln oder mit echten Daten arbeiten. Verwenden Sie stattdessen den Verbund mit einem Identitätsanbieter wie AWS IAM Identity Center.

Wenn Sie einen IAM-Benutzer verwenden, um Ihren Code auszuführen, authentifiziert sich das SDK oder das Tool in Ihrer Entwicklungsumgebung mithilfe langfristiger IAM-Benutzeranmeldeinformationen in der gemeinsam genutzten Datei. AWS `credentials` Lesen Sie das Thema Bewährte Sicherheitsmethoden in IAM und wechseln Sie so bald wie möglich zu IAM Identity Center oder anderen temporären Anmeldeinformationen.

Wichtige Warnhinweise und Richtlinien für Anmeldeinformationen

Warnhinweise für Anmeldeinformationen

- Verwenden Sie NICHT die Root-Anmeldeinformationen Ihres Kontos, um auf Ihre AWS -Ressourcen zuzugreifen. Diese Anmeldeinformationen bieten uneingeschränkten Zugriff auf Konten und können nur schwer widerrufen werden.
- Fügen Sie KEINE tatsächlichen Zugriffsschlüssel oder Anmeldeinformationen in Ihre Anwendungsdateien ein. Wenn Sie dies tun, riskieren Sie damit, dass Ihre Kontodaten versehentlich offengelegt werden, falls Sie z. B. das Projekt in ein öffentliches Repository hochladen.
- Fügen Sie KEINE Dateien in Ihrem Projektbereich hinzu, die Anmeldeinformationen enthalten.
- Beachten Sie, dass alle in der gemeinsam genutzten AWS `credentials` Datei gespeicherten Anmeldeinformationen im Klartext gespeichert werden.

Zusätzliche Hinweise zur sicheren Verwaltung von Anmeldeinformationen

Eine allgemeine Erläuterung der sicheren Verwaltung von AWS Anmeldeinformationen finden Sie unter Bewährte Methoden für die Verwaltung von AWS Zugriffsschlüsseln in der Allgemeine AWS-Referenz. Berücksichtigen Sie zusätzlich zu diesen Informationen Folgendes:

- Verwenden Sie IAM-Rollen für Aufgaben in Verbindung mit Aufgaben von Amazon Elastic Container Service (Amazon ECS).
- Verwenden Sie IAM-Rollen für Anwendungen, die auf Amazon-EC2-Instances ausgeführt werden.

Voraussetzungen: Erstellen Sie ein AWS Konto

Um einen IAM-Benutzer für den Zugriff auf AWS Dienste zu verwenden, benötigen Sie ein AWS Konto und AWS Anmeldeinformationen.

1. Erstellen Sie ein Konto.

Informationen zum Erstellen eines AWS Kontos finden Sie unter Erste Schritte: Sind Sie ein Erstbenutzer? AWS im AWS -Kontenverwaltung Referenzhandbuch.

2. Erstellen Sie einen Administratorbenutzer.

   Vermeiden Sie es, Ihr Root-Benutzerkonto (das erste Konto, das Sie erstellen) für den Zugriff auf die Managementkonsole und Services zu verwenden. Erstellen Sie stattdessen ein Administratorkonto, wie unter Erstellen eines Administratorbenutzers im IAM-Benutzerhandbuch beschrieben.

   Nachdem Sie das Administratorkonto erstellt und die Anmeldeinformationen aufgezeichnet haben, müssen Sie sich von Ihrem Root-Benutzerkonto abmelden und mit dem Administratorkonto wieder anmelden.

Keines dieser Konten ist für die Entwicklung AWS oder Ausführung von Anwendungen geeignet AWS. Es hat sich bewährt, Benutzer, Berechtigungssätze oder Servicerollen zu erstellen, die für diese Aufgaben geeignet sind. Weitere Informationen finden Sie unter Anwenden von Berechtigungen mit geringsten Berechtigungen im IAM-Benutzerhandbuch.

Schritt 1: Erstellen Ihres IAM-Benutzers

- Erstellen Sie Ihren IAM-Benutzer, indem Sie das Verfahren Erstellen von IAM-Benutzern (Konsole) im IAM-Benutzerhandbuch befolgen. Gehen Sie beim Erstellen Ihres IAM-Benutzers wie folgt vor:
  - Wir empfehlen Ihnen, Benutzerzugriff auf die AWS-Managementkonsole bereitstellen auszuwählen. Auf diese Weise können Sie den Code, den Sie gerade ausführen, in einer visuellen Umgebung anzeigen AWS-Services , z. B. beim Überprüfen von AWS CloudTrail Diagnoseprotokollen oder beim Hochladen von Dateien in Amazon Simple Storage Service, was beim Debuggen Ihres Codes hilfreich ist.
  - Wählen Sie unter Berechtigungen festlegen — Berechtigungsoptionen die Option Richtlinien direkt anhängen aus, um festzulegen, wie Sie diesem Benutzer Berechtigungen zuweisen möchten.
    - Die meisten SDK-Tutorials zum Thema „Erste Schritte“ verwenden den Amazon-S3-Service als Beispiel. Wenn Sie Ihrer Anwendung Vollzugriff auf Amazon S3 gewähren möchten, wählen Sie die `AmazonS3FullAccess`-Richtlinie zum Anfügen an diesen Benutzer aus.

- Sie können die optionalen Schritte dieses Verfahrens zum Festlegen von Berechtigungsgrenzen oder Tags ignorieren.

Schritt 2: Abrufen Ihrer Zugriffsschlüssel

1. Wählen Sie im Navigationsbereich der IAM-Konsole Benutzer und dann den **`User name`** des Benutzers aus, den Sie zuvor erstellt haben.
2. Wählen Sie auf der Seite des Benutzers die Seite Sicherheitsanmeldeinformationen aus. Wählen Sie dann unter Zugriffsschlüssel die Option Zugriffsschlüssel erstellen aus.
3. Wählen Sie für Schritt 1 „Zugriffsschlüssel erstellen“ entweder Befehlszeilenschnittstelle (CLI) oder Lokaler Code aus. Beide Optionen generieren denselben Schlüsseltyp, der sowohl mit dem als auch mit dem AWS CLI verwendet werden kann SDKs.
4. Geben Sie für Zugriffsschlüssel erstellen – Schritt 2 ein optionales Tag ein und wählen Sie Weiter aus.
5. Wählen Sie unter Zugriffsschlüssel erstellen – Schritt 3 die Option CSV-Datei herunterladen aus, um eine `.csv`-Datei mit dem Zugriffsschlüssel und dem geheimen Zugriffsschlüssel Ihres IAM-Benutzers zu speichern. Sie benötigen diese Informationen später wieder.

   > ⚠ Warning
   >
   > Verwenden Sie geeignete Sicherheitsmaßnahmen, um diese Anmeldeinformationen zu schützen.

6. Wählen Sie Done (Fertig).

Schritt 3: Aktualisieren Sie die gemeinsam genutzte **`credentials`** Datei

1. Erstellen oder öffnen Sie die freigegebene AWS `credentials`-Datei. Diese Datei befindet sich in Linux- und macOS-Systemen im Pfad `~/.aws/credentials` und unter Windows im Pfad `%USERPROFILE%\.aws\credentials`. Weitere Informationen finden Sie unter Speicherort der Anmeldeinformationsdateien.
2. Fügen Sie der freigegebenen `credentials`-Datei den folgenden Text hinzu. Ersetzen Sie den Beispiel-ID-Wert und den Beispielschlüsselwert durch die Werte in der `.csv` Datei, die Sie zuvor heruntergeladen haben.

```
[default]
aws_access_key_id = AKIAIOSFODNN7EXAMPLE
aws_secret_access_key = wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
```

3. Speichern Sie die Datei.

Die gemeinsam genutzte `credentials` Datei ist die gängigste Methode zum Speichern von Anmeldeinformationen. Diese können auch als Umgebungsvariablen festgelegt werden. Informationen zu Namen von Umgebungsvariablen finden Sie unterAWS Zugriffstasten. Dies ist eine Möglichkeit, Ihnen den Einstieg zu erleichtern. Wir empfehlen Ihnen jedoch, so bald wie möglich auf IAM Identity Center oder andere temporäre Anmeldeinformationen umzusteigen. Denken Sie nach der Umstellung auf die Verwendung langfristiger Anmeldeinformationen daran, diese Anmeldeinformationen aus der gemeinsam genutzten `credentials` Datei zu löschen.

Verwenden von IAM-Rollen zur Authentifizierung von Anwendungen, die auf Amazon bereitgestellt werden EC2

Dieses Beispiel behandelt die Einrichtung einer AWS Identity and Access Management Rolle mit Amazon S3 S3-Zugriff zur Verwendung in Ihrer Anwendung, die auf einer Amazon Elastic Compute Cloud-Instance bereitgestellt wird.

Um Ihre AWS SDK-Anwendung auf einer Amazon Elastic Compute Cloud-Instance auszuführen, erstellen Sie eine IAM-Rolle und gewähren Sie dann Ihrer EC2 Amazon-Instance Zugriff auf diese Rolle. Weitere Informationen finden Sie unter IAM-Rollen für Amazon EC2 im EC2 Amazon-Benutzerhandbuch.

Erstellen einer IAM-Rolle

Die AWS SDK-Anwendung, die Sie entwickeln, greift wahrscheinlich auf mindestens eine AWS-Service zu, um Aktionen auszuführen. Erstellen Sie eine IAM-Rolle, die die erforderlichen Berechtigungen gewährt, die für die Ausführung Ihrer Anwendung erforderlich sind.

Mit diesem Verfahren wird beispielsweise eine Rolle erstellt, die nur Lesezugriff auf Amazon S3 gewährt. Viele der AWS SDK-Leitfäden enthalten Tutorials für „Erste Schritte“, die aus Amazon S3 stammen.

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im Navigationsbereich Rollen und anschließend Rolle erstellen aus.
3. Wählen Sie für Vertrauenswürdige Entität auswählen unter Vertrauenswürdiger Entitätstyp die Option AWS-Service.
4. Wählen Sie unter Anwendungsfall die Option Amazon EC2 und dann Weiter aus.
5. Aktivieren Sie für Berechtigungen hinzufügen das Kontrollkästchen für Amazon S3 Read Only Access aus der Richtlinienliste und wählen Sie dann Weiter aus.
6. Geben Sie einen Namen für die Rolle ein und wählen Sie dann Rolle erstellen aus. Merken Sie sich diesen Namen, da Sie ihn benötigen, wenn Sie Ihre EC2 Amazon-Instance erstellen.

Starten Sie eine EC2 Amazon-Instance und geben Sie Ihre IAM-Rolle an

Gehen Sie wie folgt vor, um eine EC2 Amazon-Instance mithilfe Ihrer IAM-Rolle zu erstellen und zu starten:

- Folgen Sie Quickly launch an instance im EC2 Amazon-Benutzerhandbuch. Gehen Sie vor dem letzten Einreichungsschritt jedoch auch wie folgt vor:
  - Wählen Sie unter Erweiterte Details für das IAM-Instanzprofil die Rolle aus, die Sie im vorherigen Schritt erstellt haben.

Mit diesem IAM- und EC2 Amazon-Setup können Sie Ihre Anwendung auf der EC2 Amazon-Instance bereitstellen und Ihre Anwendung erhält Lesezugriff auf den Amazon S3-Service.

Connect zur EC2 Instanz her

Connect zur EC2 Amazon-Instance her, sodass Sie Ihre Anwendung darauf übertragen und die Anwendung dann ausführen können. Sie benötigen die Datei, die den privaten Teil des Schlüsselpaars enthält, das Sie unter key pair (Anmeldung) verwendet haben, als Sie Ihre Instance erstellt haben, also die PEM-Datei.

Sie können dies tun, indem Sie den Anweisungen für Ihren Instance-Typ folgen: Connect zu Ihrer Linux-Instance her oder Stellen Sie eine Verbindung zu Ihrer Windows-Instance her. Wenn Sie eine Verbindung herstellen, tun Sie dies so, dass Sie Dateien von Ihrem Entwicklungscomputer auf Ihre Instance übertragen können.

> **Note**
>
> Auf einem Linux- oder macOS-Terminal können Sie den Befehl Secure Copy verwenden, um Ihre Anwendung zu kopieren. Zur Verwendung `scp` mit einem key pair können Sie den folgenden Befehl verwenden:`scp -i path/to/key file/to/copy ec2-user@ec2-xx-xx-xxx-xxx.compute.amazonaws.com:~`.
> Weitere Informationen für Windows finden Sie unter Dateien auf Windows-Instanzen übertragen.

Wenn Sie ein AWS Toolkit verwenden, können Sie häufig auch mithilfe des Toolkits eine Verbindung zu der Instanz herstellen. Weitere Informationen finden Sie in der spezifischen Bedienungsanleitung für das von Ihnen verwendete Toolkit.

Führen Sie Ihre Anwendung auf der Instanz aus EC2

1. Kopieren Sie Ihre Anwendungsdateien von Ihrem lokalen Laufwerk auf Ihre EC2 Amazon-Instance.
2. Starten Sie die Anwendung und stellen Sie sicher, dass sie mit den gleichen Ergebnissen wie auf Ihrem Entwicklungscomputer ausgeführt wird.
3. (Optional) Stellen Sie sicher, dass die Anwendung die von der IAM-Rolle bereitgestellten Anmeldeinformationen verwendet.

    a. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die EC2 Amazon-Konsole unter https://console.aws.amazon.com/ec2/.
    b. Wählen Sie die Instance aus.
    c. Wählen Sie Aktionen, Sicherheit und anschließend IAM-Rolle ändern aus.
    d. Trennen Sie für die IAM-Rolle die IAM-Rolle, indem Sie „Keine IAM-Rolle“ wählen.
    e. Wählen Sie IAM-Rolle aktualisieren.
    f. Führen Sie die Anwendung erneut aus und vergewissern Sie sich, dass sie einen Autorisierungsfehler zurückgibt.

Verwenden des TIP-Plugins für den Zugriff AWS-Services

Trusted Identity Propagation (TIP) ist eine Funktion AWS IAM Identity Center , die es Administratoren ermöglicht, Berechtigungen auf der Grundlage von Benutzerattributen wie Gruppenzuordnungen

AWS-Services zu gewähren. Bei Trusted Identity Propagation wird einer IAM-Rolle ein Identitätskontext hinzugefügt, um den Benutzer zu identifizieren, der Zugriff auf AWS Ressourcen anfordert. Dieser Kontext wird an andere weitergegeben. AWS-Services

Der Identitätskontext umfasst Informationen, AWS-Services anhand derer Autorisierungsentscheidungen getroffen werden, wenn sie Zugriffsanfragen erhalten. Zu diesen Informationen gehören Metadaten, mit denen der Anforderer (z. B. ein IAM Identity Center-Benutzer), der AWS-Service Zugriff angefordert wird (z. B. Amazon Redshift) und der Zugriffsumfang (z. B. schreibgeschützter Zugriff) identifiziert werden. Der Empfänger AWS-Service verwendet diesen Kontext und alle dem Benutzer zugewiesenen Berechtigungen, um den Zugriff auf seine Ressourcen zu autorisieren. Weitere Informationen finden Sie in der Übersicht über die Verbreitung vertrauenswürdiger Identitäten im AWS IAM Identity Center Benutzerhandbuch.

Das TIP-Plugin kann zusammen mit AWS-Services diesem Plugin verwendet werden, das die Verbreitung vertrauenswürdiger Identitäten unterstützt. Einen Referenzanwendungsfall finden Sie unter Konfiguration einer Amazon Q Business-Anwendung mithilfe von Amazon Q AWS IAM Identity Center im Amazon Q Business-Benutzerhandbuch.

> **Note**
>
> Wenn Sie Amazon Q Business verwenden, finden Sie unter Konfiguration einer Amazon Q Business-Anwendung mithilfe AWS IAM Identity Center von servicespezifischen Anweisungen.

Voraussetzungen für die Verwendung des TIP-Plug-ins

Die folgenden Ressourcen sind erforderlich, damit das Plugin funktioniert:

1. Sie müssen entweder das AWS SDK für Java oder das verwenden AWS SDK für JavaScript.
2. Stellen Sie sicher, dass der Dienst, den Sie verwenden, die Verbreitung vertrauenswürdiger Identitäten unterstützt.

   Weitere Informationen finden Sie in der Spalte Aktiviert die Verbreitung vertrauenswürdiger Identitäten über IAM Identity Center in der Tabelle AWS verwaltete Anwendungen, die in IAM Identity Center integriert sind, im AWS IAM Identity Center Benutzerhandbuch.
3. Aktivieren Sie IAM Identity Center und die Verbreitung vertrauenswürdiger Identitäten.

Weitere Informationen zu den Voraussetzungen und Überlegungen zu TIP finden Sie im AWS IAM Identity Center Benutzerhandbuch.

4. Sie müssen über eine Identity-Center-integrated Bewerbung verfügen.

   Weitere Informationen finden Sie AWS im AWS IAM Identity Center Benutzerhandbuch unter Verwaltete Anwendungen oder Vom Kunden verwaltete Anwendungen.

5. Sie müssen einen vertrauenswürdigen Token-Aussteller (TTI) einrichten und Ihren Service mit dem IAM Identity Center verbinden.

   Weitere Informationen finden Sie im Benutzerhandbuch unter Voraussetzungen für vertrauenswürdige Token-Aussteller und Aufgaben für die Einrichtung eines vertrauenswürdigen Token-Ausstellers.AWS IAM Identity Center

Um das TIP-Plugin in Ihrem Code zu verwenden

1. Erstellen Sie eine Instanz des Plugins zur Verbreitung vertrauenswürdiger Identitäten.
2. Erstellen Sie eine Service-Client-Instanz für die Interaktion mit Ihrem AWS-Service und passen Sie den Service-Client an, indem Sie das Trusted Identity Propagation-Plugin hinzufügen.

Das TIP-Plugin verwendet die folgenden Eingabeparameter:

- **`webTokenProvider`**: Eine Funktion, die der Kunde implementiert, um ein OpenID-Token von seinem externen Identitätsanbieter zu erhalten.
- **`accessRoleArn`**: Der ARN der IAM-Rolle, den das Plugin mit dem Identitätskontext des Benutzers annehmen soll, um die identitätserweiterten Anmeldeinformationen abzurufen.
- **`applicationArn`**: Die eindeutige Kennungszeichenfolge für den Client oder die Anwendung. Dieser Wert ist ein Anwendungs-ARN, für den OAuth Grants konfiguriert sind.
- **`ssoOidcClient`**: (Optional) Ein SSO-OIDC-Client, z. B. `SsoOidcClient`für Java oder `client-sso-oidc`für JavaScript, mit kundenspezifischen Konfigurationen. Falls nicht angegeben, wird ein OIDC-Client mithilfe von `applicationRoleArn` instanziiert und verwendet.
- **`stsClient`**: (Optional) Ein AWS STS Client mit kundendefinierten Konfigurationen, der verwendet wird, um den Identitätskontext des Benutzers `accessRoleArn` zu berücksichtigen. Falls nicht angegeben, `applicationRoleArn` wird ein AWS STS verwendender Client instanziiert und verwendet.

- **`applicationRoleArn`**: (Optional) Der ARN der IAM-Rolle, mit dem angenommen werden soll, `AssumeRoleWithWebIdentity` damit der OIDC und die AWS STS Clients gebootet werden können.
  - Wenn nicht angegeben, müssen sowohl der als auch der `ssoOidcClient` Parameter angegeben werden. `stsClient`
  - Falls angegeben, `applicationRoleArn` kann es nicht derselbe Wert wie der `accessRoleArn` Parameter sein. `applicationRoleArn`wird verwendet, um den STSClient zu erstellen, der verwendet wird, um AccessRole zu übernehmen. Wenn dieselbe Rolle für beide `applicationRole` und verwendet wird, würde das bedeuten`accessRole`, eine Rolle zu verwenden, um sich selbst anzunehmen (Übernahme der eigenen Rolle), wovon abgeraten wird. AWS Weitere Einzelheiten finden Sie in der Ankündigung.

Überlegungen zu **`ssoOidcClient`**`stsClient`, und **`applicationRoleArn`** Parametern

Beachten Sie bei der Konfiguration des TIP-Plug-ins die folgenden Berechtigungsanforderungen, je nachdem, welche Parameter Sie angeben:

- Wenn Sie angeben `ssoOidcClient` und`stsClient`:
  - Die Anmeldeinformationen auf der `ssoOidcClient` sollten `oauth:CreateTokenWithIAM` berechtigt sein, Identity Center anzurufen, um den Identity Center-spezifischen Benutzerkontext abzurufen.
  - Die Anmeldeinformationen `stsClient` sollten aktiviert sein und `sts:AssumeRole` die `sts:SetContext` Berechtigungen müssen aktiviert sein`accessRole`. `accessRole`muss außerdem mit einer Vertrauensstellung konfiguriert werden, bei der die Anmeldeinformationen aktiviert sind`stsClient`.
- Wenn Sie Folgendes bereitstellen`applicationRoleArn`:
  - `applicationRole`sollte über die `oauth:CreateTokenWithIAM` `sts:SetContext` Berechtigungen `sts:AssumeRole` und für die erforderlichen Ressourcen (IdC-Instanz`accessRole`) verfügen, da sie zum Erstellen von OIDC- und STS-Clients verwendet wird.
  - `applicationRole`sollte eine Vertrauensbeziehung mit dem Identitätsanbieter haben, der für die Generierung verwendet wird`webToken`, da dieser verwendet `webToken` wird, um die ApplicationRole über den AssumeRoleWithWebIdentityAufruf durch das Plugin zu übernehmen.

ApplicationRole Beispielkonfiguration:

Vertrauensrichtlinie mit Web-Token-Anbieter:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Federated": "arn:aws:iam::ACCOUNT_ID:oidc-provider/
IDENTITY_PROVIDER_URL"
            },
            "Action": "sts:AssumeRoleWithWebIdentity",
            "Condition": {
                "StringEquals": {
                    "IDENTITY_PROVIDER_URL:aud": "CLIENT_ID_TO_BE_TRUSTED"
                }
            }
        }
    ]
}
```

Genehmigungsrichtlinie:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sts:AssumeRole",
                "sts:SetContext"
            ],
            "Resource": [
                "accessRoleArn"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "sso-oauth:CreateTokenWithIAM"
            ],
```

```
            "Resource": [
                "*"
            ]
        }
    ]
}
```

Codebeispiele mit TIP

Die folgenden Beispiele zeigen, wie Sie das TIP-Plugin mit dem AWS SDK für Java oder dem in Ihrem Code implementieren AWS SDK für JavaScript.

Java

Um das TIP-Plugin in Ihrem AWS SDK für Java Projekt zu verwenden, müssen Sie es als Abhängigkeit in der `pom.xml` Datei Ihres Projekts deklarieren.

```
<dependency>
<groupId>software.amazon.awsidentity.trustedIdentityPropagation</groupId>
<artifactId>aws-sdk-java-trustedIdentityPropagation-java-plugin</artifactId>
   <version>2.0.0</version>
</dependency>
```

Fügen Sie in Ihrem Quellcode die erforderliche Paketanweisung für ein`software.amazon.awssdk.trustedidentitypropagation`.

Die folgenden Beispiele zeigen zwei Möglichkeiten, eine Instanz des Trusted Identity Propagation Plug-ins zu erstellen und sie einem Service-Client hinzuzufügen. In beiden Beispielen wird Amazon S3 als Service und `S3AccessGrantsPlugin` zur Verwaltung benutzerspezifischer Berechtigungen verwendet. Sie können jedoch auch auf alle Anwendungen angewendet werden, AWS-Service die Trusted Identity Propagation (TIP) unterstützen.

> **Note**
>
> Für diese Beispiele müssen Sie die benutzerspezifischen Berechtigungen von S3 Access Grants einrichten. Weitere Informationen finden Sie in der Dokumentation zu S3 Access Grants.

Option 1: OIDC- und STS-Clients erstellen und übergeben

```
SsoOidcClient oidcClient = SsoOidcClient.builder()
    .region(Region.US_EAST_1)
    .credentialsProvider(credentialsProvider).build();

StsClient stsClient = StsClient.builder()
    .region(Region.US_EAST_1)
    .credentialsProvider(credentialsProvider).build();

TrustedIdentityPropagationPlugin trustedIdentityPropagationPlugin =
 TrustedIdentityPropagationPlugin.builder()
        .webTokenProvider(() -> webToken)
        .applicationArn(idcApplicationArn)
        .accessRoleArn(accessRoleArn)
        .ssoOidcClient(oidcClient)
        .stsClient(stsClient)
        .build();

S3AccessGrantsPlugin accessGrantsPlugin = S3AccessGrantsPlugin.builder()
        .build();

S3Client s3Client =
        S3Client.builder().region(Region.US_EAST_1)
                .crossRegionAccessEnabled(true)
                .addPlugin(trustedIdentityPropagationPlugin)
                .addPlugin(accessGrantsPlugin)
                .build();

final var resp = s3Client.getObject(GetObjectRequest.builder()
        .key("path/to/object/fileName")
        .bucket("bucketName")
        .build());
```

Option 2: Übergeben applicationRoleArn und verschieben Sie die Client-Erstellung an das Plugin

```
TrustedIdentityPropagationPlugin trustedIdentityPropagationPlugin =
 TrustedIdentityPropagationPlugin.builder()
        .webTokenProvider(() -> webToken)
        .applicationArn(idcApplicationArn)
        .accessRoleArn(accessRoleArn)
        .applicationRoleArn(applicationRoleArn)
        .build();

S3AccessGrantsPlugin accessGrantsPlugin = S3AccessGrantsPlugin.builder()
```

```
        .build();

S3Client s3Client =
        S3Client.builder().region(Region.US_EAST_1)
                .crossRegionAccessEnabled(true)
                .addPlugin(trustedIdentityPropagationPlugin)
                .addPlugin(accessGrantsPlugin)
                .build();

final var resp = s3Client.getObject(GetObjectRequest.builder()
        .key("path/to/object/fileName")
        .bucket("bucketName")
        .build());
```

Weitere Informationen und Quellen finden Sie trusted-identity-propagation-javaunter GitHub.

JavaScript

Führen Sie den folgenden Befehl aus, um das TIP-Authentifizierungs-Plugin-Paket in Ihrem AWS SDK für JavaScript Projekt zu installieren:

```
$  npm i @aws-sdk-extension/trusted-identity-propagation
```

Das Finale `package.json` sollte eine Abhängigkeit enthalten, die der folgenden ähnelt:

```
  "dependencies": {
"@aws-sdk-extension/trusted-identity-propagation": "^2.0.0"
  },
```

Importieren Sie die erforderliche `TrustedIdentityPropagationExtension` Abhängigkeit in Ihren Quellcode.

Die folgenden Beispiele zeigen zwei Möglichkeiten, eine Instanz des Trusted Identity Propagation Plug-ins zu erstellen und sie einem Service-Client hinzuzufügen. Beide Beispiele verwenden Amazon S3 als Service und Amazon S3 Access Grants zur Verwaltung benutzerspezifischer Berechtigungen, können aber auch auf alle angewendet werden, AWS-Service die Trusted Identity Propagation (TIP) unterstützen.

> **Note**
>
> Für diese Beispiele müssen Sie die benutzerspezifischen Berechtigungen von Amazon S3 Access Grants einrichten. Weitere Informationen finden Sie in der Dokumentation zu Amazon S3 Access Grants.

Option 1: OIDC- und STS-Clients erstellen und übergeben

```
import { S3Client, GetObjectCommand } from "@aws-sdk/client-s3";
import { S3ControlClient, GetDataAccessCommand } from "@aws-sdk/client-s3-control";
import { TrustedIdentityPropagationExtension } from "@aws-sdk-extension/trusted-
identity-propagation";

const s3ControlClient = new S3ControlClient({
    region: "us-east-1",
    extensions: [
        TrustedIdentityPropagationExtension.create({
            webTokenProvider: async () => {
                return 'ID_TOKEN_FROM_YOUR_IDENTITY_PROVIDER';
            },
            ssoOidcClient: customOidcClient,
            stsClient: customStsClient,
            accessRoleArn: accessRoleArn,
            applicationArn: applicationArn,
        }),
    ],
});

const getDataAccessParams = {
  Target: "S3_URI_PATH",
  Permission: "READ",
  AccountId: ACCOUNT_ID,
  InstanceArn: S3_ACCESS_GRANTS_ARN,
  TargetType: "Object",
};

try {
  const command = new GetDataAccessCommand(getDataAccessParams);
  const response = await s3ControlClient.send(command);

  const credentials = response.Credentials;
```

```
  // Create a new S3 client with the temporary credentials
  const temporaryS3Client = new S3Client({
    region: "us-east-1",
    credentials: {
      accessKeyId: credentials.AccessKeyId,
      secretAccessKey: credentials.SecretAccessKey,
      sessionToken: credentials.SessionToken,
    },
  });

  // Use the temporary S3 client to perform the operation
  const s3Params = {
    Bucket: "BUCKET_NAME",
    Key: "S3_OBJECT_KEY",
  };
  const getObjectCommand = new GetObjectCommand(s3Params);
  const s3Object = await temporaryS3Client.send(getObjectCommand);

  const fileContent = await s3Object.Body.transformToString();

  // Process the S3 object data
  console.log("Successfully retrieved S3 object:", fileContent);
} catch (error) {
  console.error("Error accessing S3 data:", error);
}
```

Option 2: Übergeben applicationRoleArn und verschieben Sie die Client-Erstellung an das Plugin

```
import { S3Client, GetObjectCommand } from "@aws-sdk/client-s3";
import { S3ControlClient, GetDataAccessCommand } from "@aws-sdk/client-s3-control";
import { TrustedIdentityPropagationExtension } from "@aws-sdk-extension/trusted-
identity-propagation";

const s3ControlClient = new S3ControlClient({
    region: "us-east-1",
    extensions: [
        TrustedIdentityPropagationExtension.create({
            webTokenProvider: async () => {
                return 'ID_TOKEN_FROM_YOUR_IDENTITY_PROVIDER';
            },
            accessRoleArn: accessRoleArn,
            applicationRoleArn: applicationRoleArn,
```

```
            applicationArn: applicationArn,
        }),
    ],
});

// Same S3 AccessGrants workflow as Option 1
const getDataAccessParams = {
  Target: "S3_URI_PATH",
  Permission: "READ",
  AccountId: ACCOUNT_ID,
  InstanceArn: S3_ACCESS_GRANTS_ARN,
  TargetType: "Object",
};

try {
  const command = new GetDataAccessCommand(getDataAccessParams);
  const response = await s3ControlClient.send(command);

  const credentials = response.Credentials;

  const temporaryS3Client = new S3Client({
    region: "us-east-1",
    credentials: {
      accessKeyId: credentials.AccessKeyId,
      secretAccessKey: credentials.SecretAccessKey,
      sessionToken: credentials.SessionToken,
    },
  });

  const s3Params = {
    Bucket: "BUCKET_NAME",
    Key: "S3_OBJECT_KEY",
  };
  const getObjectCommand = new GetObjectCommand(s3Params);
  const s3Object = await temporaryS3Client.send(getObjectCommand);

  const fileContent = await s3Object.Body.transformToString();

  console.log("Successfully retrieved S3 object:", fileContent);
} catch (error) {
  console.error("Error accessing S3 data:", error);
}
```

Weitere Informationen und Quellen finden Sie trusted-identity-propagation-jsunter GitHub.

AWS SDKs und Referenz zu den Werkzeugeinstellungen

SDKs stellen sprachspezifisch APIs für bereit. AWS-Services Sie übernehmen einige der schweren Aufgaben, die für erfolgreiche API-Aufrufe erforderlich sind, einschließlich Authentifizierung, Wiederholungsverhalten und mehr. Zu diesem Zweck SDKs verfügen sie über flexible Strategien zum Abrufen von Anmeldeinformationen für Ihre Anfragen, zur Verwaltung der Einstellungen für die einzelnen Dienste und zum Abrufen von Werten, die für globale Einstellungen verwendet werden können.

In den folgenden Abschnitten finden Sie detaillierte Informationen zu den Konfigurationseinstellungen:

- AWS SDKs und Tools standardisierte Anbieter von Anmeldeinformationen— Gängige Anbieter von Anmeldeinformationen, die für mehrere SDKs standardisiert sind.
- AWS Standardisierte Funktionen von SDKs und Tools— Gemeinsame Funktionen, die für mehrere SDKs standardisiert sind.

Serviceclients erstellen

SDKs Verwenden Sie class/object für den AWS-Services programmgesteuerten Zugriff jeweils einen Client. AWS-Service Wenn Ihre Anwendung beispielsweise auf Amazon zugreifen muss EC2, erstellt Ihre Anwendung ein EC2 Amazon-Client-Objekt als Schnittstelle zu diesem Service. Anschließend verwenden Sie den Service-Client, um Anfragen an dieses zu stellen AWS-Service. In den meisten SDKs Fällen ist ein Service-Client-Objekt unveränderlich, sodass Sie für jeden Dienst, an den Sie Anfragen stellen, und für Anfragen an denselben Dienst mit einer anderen Konfiguration einen neuen Client erstellen müssen.

Vorrang der Einstellungen

In globalen Einstellungen werden Funktionen, Anbieter von Anmeldeinformationen und andere Funktionen konfiguriert, die von den meisten unterstützt werden SDKs und weitreichende Auswirkungen auf alle haben. AWS-Services Alle SDKs haben eine Reihe von Orten (oder Quellen), die sie überprüfen, um einen Wert für globale Einstellungen zu finden. Im Folgenden wird die Rangfolge der Suchvorgänge festgelegt:

1. Jede explizite Einstellung, die im Code oder auf einem Service-Client selbst festgelegt ist, hat Vorrang vor allen anderen Einstellungen.

- Einige Einstellungen können pro Vorgang festgelegt und bei Bedarf für jeden Vorgang, den Sie aufrufen, geändert werden. Bei AWS CLI oder handelt AWS -Tools für PowerShell es sich um Parameter pro Vorgang, die Sie in der Befehlszeile eingeben. Bei einem SDK können explizite Zuweisungen die Form eines Parameters annehmen, den Sie festlegen, wenn Sie einen AWS-Service Client oder ein Konfigurationsobjekt instanziieren, oder manchmal, wenn Sie eine einzelne API aufrufen.

2. Nur Java/Kotlin: Die JVM-Systemeigenschaft für die Einstellung ist überprüft. Wenn sie gesetzt ist, wird dieser Wert zur Konfiguration des Clients verwendet.
3. Die Umgebungsvariable wird geprüft. Wenn er gesetzt ist, wird dieser Wert zur Konfiguration des Clients verwendet.
4. Das SDK überprüft die gemeinsam genutzte `credentials` Datei auf die Einstellung. Wenn sie festgelegt ist, verwendet der Client sie.
5. Die gemeinsam genutzte `config` Datei für die Einstellung. Wenn die Einstellung vorhanden ist, verwendet das SDK sie.
   - Die `AWS_PROFILE` Umgebungsvariable oder die `aws.profile` JVM-Systemeigenschaft kann verwendet werden, um anzugeben, welches Profil das SDK lädt.
6. Jeder vom SDK-Quellcode selbst bereitgestellte Standardwert wird zuletzt verwendet.

> **Note**
>
> Bei einigen SDKs AND-Tools wird die Prüfung möglicherweise in einer anderen Reihenfolge durchgeführt. Einige SDKs AND-Tools unterstützen auch andere Methoden zum Speichern und Abrufen von Parametern. Beispielsweise AWS SDK für .NET unterstützt das eine zusätzliche Quelle namens SDK Store. Weitere Informationen zu Anbietern, die nur für ein SDK oder Tool verfügbar sind, finden Sie in der spezifischen Anleitung für das von Ihnen verwendete SDK oder Tool.

Die Reihenfolge bestimmt, welche Methoden Vorrang haben und andere überschreiben. Wenn Sie beispielsweise ein Profil in der gemeinsam genutzten `config` Datei einrichten, wird es erst gefunden und verwendet, nachdem das SDK oder Tool zuerst die anderen Orte überprüft hat. Das heißt, wenn Sie eine Einstellung in die `credentials` Datei einfügen, wird diese anstelle der in der `config` Datei enthaltenen Einstellung verwendet. Wenn Sie eine Umgebungsvariable mit einer Einstellung und einem Wert konfigurieren, würde diese Einstellung sowohl in der als auch in der `credentials` `config` Datei außer Kraft gesetzt. Und schließlich würde eine Einstellung für die einzelne Operation

(AWS CLI Befehlszeilenparameter oder API-Parameter) oder im Code alle anderen Werte für diesen einen Befehl überschreiben.

Die Einstellungsseiten dieses Handbuchs verstehen

Auf den Seiten im Referenzabschnitt zu den Einstellungen dieses Handbuchs werden die verfügbaren Einstellungen detailliert beschrieben, die über verschiedene Mechanismen festgelegt werden können. In den folgenden Tabellen sind die Einstellungen für die Konfiguration und die Anmeldeinformationsdatei, Umgebungsvariablen und (für Java und Kotlin SDKs) die JVM-Einstellungen aufgeführt, die außerhalb Ihres Codes zur Konfiguration der Funktion verwendet werden können. Jedes verlinkte Thema in jeder Liste führt Sie zur entsprechenden Einstellungsseite.

- `Config`Liste der Dateieinstellungen
- `Credentials`Liste der Dateieinstellungen
- Liste der Umgebungsvariablen
- Liste der JVM-Systemeigenschaften

Jeder Anmeldeinformationsanbieter oder jede Funktion hat eine Seite, auf der die Einstellungen aufgeführt sind, die zur Konfiguration dieser Funktionalität verwendet werden. Für jede Einstellung können Sie den Wert oft festlegen, indem Sie die Einstellung entweder zu einer Konfigurationsdatei hinzufügen oder indem Sie eine Umgebungsvariable setzen oder (nur für Java und Kotlin), indem Sie eine JVM-Systemeigenschaft festlegen. Jede Einstellung listet alle unterstützten Methoden zum Setzen des Werts in einem Block über den Details der Beschreibung auf. Die Rangfolge ist zwar unterschiedlich, die daraus resultierende Funktionalität ist jedoch dieselbe, unabhängig davon, wie Sie sie einstellen.

Die Beschreibung enthält gegebenenfalls den Standardwert, der wirksam wird, wenn Sie nichts tun. Außerdem wird definiert, welcher Wert für diese Einstellung gültig ist.

Schauen wir uns zum Beispiel eine Einstellung auf der Komprimierung anfordern Feature-Seite an.

Die Informationen der `disable_request_compression` Beispieleinstellung dokumentieren Folgendes:

- Es gibt drei gleichwertige Möglichkeiten, die Komprimierung von Anfragen außerhalb Ihrer Codebasis zu steuern. Führen Sie dazu einen der folgenden Schritte aus:

- Stellen Sie es in Ihrer Konfigurationsdatei ein mit `disable_request_compression`
- Stellen Sie es als Umgebungsvariable ein mit `AWS_DISABLE_REQUEST_COMPRESSION`
- Oder, wenn Sie das Java- oder Kotlin-SDK verwenden, legen Sie es als JVM-Systemeigenschaft fest mit `aws.disableRequestCompression`

> **Note**
>
> Möglicherweise gibt es auch eine Möglichkeit, dieselbe Funktionalität direkt in Ihrem Code zu konfigurieren, aber diese Referenz behandelt dies nicht, da sie für jedes SDK einzigartig ist. Wenn Sie Ihre Konfiguration im Code selbst festlegen möchten, lesen Sie in Ihrem spezifischen SDK-Handbuch oder in der API-Referenz nach.

- Wenn Sie nichts tun, wird der Wert standardmäßig auf gesetzt`false`.
- Die einzigen gültigen Werte für diese boolesche Einstellung sind `true` und. `false`

Am Ende jeder Feature-Seite befindet sich eine Tabelle mit Support von AWS SDKs und Tools.

Diese Tabelle zeigt, ob Ihr SDK die auf der Seite aufgeführten Einstellungen unterstützt. Die `Supported` Spalte gibt die Unterstützungsstufe mit den folgenden Werten an:

- `Yes`— Die Einstellungen werden vom SDK in der geschriebenen Form vollständig unterstützt.
- `Partial`— Einige Einstellungen werden unterstützt oder das Verhalten weicht von der Beschreibung ab. Denn `Partial` ein zusätzlicher Hinweis weist auf die Abweichung hin.
- No— Keine der Einstellungen wird unterstützt. Dies erhebt keinen Anspruch darauf, ob dieselbe Funktionalität im Code erreicht werden könnte; es weist nur darauf hin, dass die aufgelisteten externen Konfigurationseinstellungen nicht unterstützt werden.

`Config`Liste der Dateieinstellungen

Die in der folgenden Tabelle aufgeführten Einstellungen können in der gemeinsam genutzten AWS `config` Datei zugewiesen werden. Sie sind global und betreffen alle AWS-Services. SDKs und Tools können auch spezielle Einstellungen und Umgebungsvariablen unterstützen. Informationen zu den Einstellungen und Umgebungsvariablen, die nur von einem einzelnen SDK oder Tool unterstützt werden, finden Sie in dem jeweiligen SDK- oder Toolhandbuch.

Einstellungsname	Details
account_id_endpoint_mode	Kontobasierte Endpunkte
api_versions	Allgemeine Konfigurationseinstellungen
auth_scheme_preference	Authentifizierungsschema
aws_access_key_id	AWS Zugriffstasten
aws_account_id	Kontobasierte Endpunkte
aws_secret_access_key	AWS Zugriffstasten
aws_session_token	AWS Zugriffstasten
ca_bundle	Allgemeine Konfigurationseinstellungen
credential_process	Anbieter für Prozessanmeldeinformationen
credential_source	Nehmen Sie die Rolle des Anbieters von Anmeldeinformationen an
defaults_mode	Standardeinstellungen für intelligente Konfigurationen
disable_host_prefix_injection	Host-Präfix-Injektion

Einstellungsname	Details
disable_request_compression	Komprimierung anfordern
duration_seconds	Nehmen Sie die Rolle des Anbieters von Anmeldeinformationen an
ec2_metadata_service_endpoint	Anbieter von IMDS-Anmeldeinformationen
ec2_metadata_service_endpoint_mode	Anbieter von IMDS-Anmeldeinformationen
ec2_metadata_v1_disabled	Anbieter von IMDS-Anmeldeinformationen
endpoint_discovery_enabled	Erkennung von Endpunkten
endpoint_url	Servicespezifische Endpunkte
external_id	Nehmen Sie die Rolle des Anbieters von Anmeldeinformationen an
ignore_configured_endpoint_urls	Dienstspezifische Endpunkte
max_attempts	Verhalten wiederholen

Einstellungsname	Details	
metadata_service_num_attempts	EC2 Amazon-Instanz-Metadaten	
metadata_service_timeout	EC2 Amazon-Instanz-Metadaten	
mfa_serial	Nehmen Sie die Rolle des Anbieters von Anmeldeinformationen an	
output	Allgemeine Konfigurationseinstellungen	
parameter_validation	Allgemeine Konfigurationseinstellungen	
region	AWS-Region	
request_checksum_calculation	Schutz der Datenintegrität für Amazon S3	
request_min_compression_size_bytes	Komprimierung anfordern	
response_checksum_validation	Schutz der Datenintegrität für Amazon S3	
retry_mode	Verhalten wiederholen	
role_arn	Nehmen Sie die Rolle des Anbieters von Anmeldeinformationen an	

Einstellungsname	Details	
`role_session_name`	Nehmen Sie die Rolle des Anbieters von Anmeldeinformationen an	
`s3_disable_express_session_auth`	S3 Express One Zone-Sitzungsauthentifizierung	
`s3_disable_multiregion_access_points`	Mulitiregionale Amazon-S3-Zugriffspunkte	
`s3_use_arn_region`	Amazon-S3-Zugriffspunkte	
`sdk_ua_app_id`	Application ID	
`sigv4a_signing_region_set`	Authentifizierungsschema	
`source_profile`	Nehmen Sie die Rolle des Anbieters von Anmeldeinformationen an	
`sso_account_id`	Anbieter von IAM Identity Center-Anmeldeinformationen	
`sso_region`	Anbieter von IAM Identity Center-Anmeldeinformationen	
`sso_registration_scopes`	Anbieter von IAM Identity Center-Anmeldeinformationen	
`sso_role_name`	Anbieter von IAM Identity Center-Anmeldeinformationen	

Einstellungsname	Details	
`sso_start_url`	Anbieter von IAM Identity Center-Anmeldeinformationen	
`sts_regional_endpoints`	AWS STS Regionale Endpunkte	
`use_dualstack_endpoint`	Dual-Stack- und FIPS-Endpunkte	
`use_fips_endpoint`	Dual-Stack- und FIPS-Endpunkte	
`web_identity_token_file`	Nehmen Sie die Rolle des Anbieters von Anmeldeinformationen an	

`Credentials`Liste der Dateieinstellungen

Die in der folgenden Tabelle aufgeführten Einstellungen können in der gemeinsam genutzten AWS `credentials` Datei zugewiesen werden. Sie sind global und betreffen alle AWS-Services. SDKs und Tools können auch spezielle Einstellungen und Umgebungsvariablen unterstützen. Informationen zu den Einstellungen und Umgebungsvariablen, die nur von einem einzelnen SDK oder Tool unterstützt werden, finden Sie in dem jeweiligen SDK- oder Toolhandbuch.

Einstellungsname	Details	
`aws_access_key_id`	AWS Zugriffstasten	
`aws_secret_access_key`	AWS Zugriffstasten	
`aws_session_token`	AWS Zugriffstasten	

Liste der Umgebungsvariablen

Die von den meisten unterstützten Umgebungsvariablen SDKs sind in der folgenden Tabelle aufgeführt. Sie sind global und betreffen alle AWS-Services. SDKs und Tools können auch spezielle Einstellungen und Umgebungsvariablen unterstützen. Informationen zu den Einstellungen und Umgebungsvariablen, die nur von einem einzelnen SDK oder Tool unterstützt werden, finden Sie in dem jeweiligen SDK- oder Toolhandbuch.

Einstellungsname	Details	
`AWS_ACCESS_KEY_ID`	AWS Zugriffstasten	
`AWS_ACCOUNT_ID`	Kontobasierte Endpunkte	
`AWS_ACCOUNT_ID_ENDPOINT_MODE`	Kontobasierte Endpunkte	
`AWS_AUTH_SCHEME_PREFERENCE`	Authentifizierungsschema	
`AWS_CA_BUNDLE`	Allgemeine Konfigurationseinstellungen	
`AWS_CONFIG_FILE`	Suchen und Ändern des Speicherorts der geteilten `credentials` Dateien `config`AWS SDKs und Tools	
`AWS_CONTAINER_AUTHORIZATION_TOKEN`	Anbieter von Container-Anmeldeinformationen	
`AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE`	Anbieter von Container-Anmeldeinformationen	

Einstellungsname	Details	
AWS_CONTAINER_CREDENTIALS_FULL_URI	Anbieter von Container-Anmeldeinformationen	
AWS_CONTAINER_CREDENTIALS_RELATIVE_URI	Anbieter von Container-Anmeldeinformationen	
AWS_DEFAULTS_MODE	Standardeinstellungen für intelligente Konfigurationen	
AWS_DISABLE_HOST_PREFIX_INJECTION	Host-Präfix-Injektion	
AWS_DISABLE_REQUEST_COMPRESSION	Komprimierung anfordern	
AWS_EC2_METADATA_DISABLED	Anbieter von IMDS-Anmeldeinformationen	
AWS_EC2_METADATA_SERVICE_ENDPOINT	Anbieter von IMDS-Anmeldeinformationen	
AWS_EC2_METADATA_SERVICE_ENDPOINT_MODE	Anbieter von IMDS-Anmeldeinformationen	

Einstellungsname	Details
AWS_EC2_METADATA_V1_DISABLED	Anbieter von IMDS-Anmeldeinformationen
AWS_ENABLE_ENDPOINT_DISCOVERY	Erkennung von Endpunkten
AWS_ENDPOINT_URL	Servicespezifische Endpunkte
AWS_ENDPOINT_URL_<SERVICE>	Servicespezifische Endpunkte
AWS_IGNORE_CONFIGURED_ENDPOINT_URLS	Servicespezifische Endpunkte
AWS_MAX_ATTEMPTS	Verhalten wiederholen
AWS_METADATA_SERVICE_NUM_ATTEMPTS	EC2 Amazon-Instanz-Metadaten
AWS_METADATA_SERVICE_TIMEOUT	EC2 Amazon-Instanz-Metadaten
AWS_PROFILE	Verwenden von geteilten `credentials` Dateien `config` und Dateien zur globalen Konfiguration von AWS SDKs Tools
AWS_REGION	AWS-Region

Einstellungsname	Details
AWS_REQUEST_CHECKSUM_CALCULATION	Schutz der Datenintegrität für Amazon S3
AWS_REQUEST_MIN_COMPRESSION_SIZE_BYTES	Komprimierung anfordern
AWS_RESPONSE_CHECKSUM_VALIDATION	Schutz der Datenintegrität für Amazon S3
AWS_RETRY_MODE	Verhalten wiederholen
AWS_ROLE_ARN	Nehmen Sie die Rolle des Anbieters von Anmeldeinformationen an
AWS_ROLE_SESSION_NAME	Nehmen Sie die Rolle des Anbieters von Anmeldeinformationen an
AWS_S3_DISABLE_EXPRESS_SESSION_AUTH	S3 Express One Zone-Sitzungsauthentifizierung
AWS_S3_DISABLE_MULTIREGION_ACCESS_POINTS	Mulitiregionale Amazon-S3-Zugriffspunkte
AWS_S3_USE_ARN_REGION	Amazon-S3-Zugriffspunkte
AWS_SDK_UA_APP_ID	Application ID

Einstellungsname	Details
AWS_SECRET_ACCESS_KEY	AWS Zugriffstasten
AWS_SESSION_TOKEN	AWS Zugriffstasten
AWS_SHARED_CREDENTIALS_FILE	Suchen und Ändern des Speicherorts der geteilten `credentials` Dateien `config`AWS SDKs und Tools
AWS_SIGV4A_SIGNING_REGION_SET	Authentifizierungsschema
AWS_STS_REGIONAL_ENDPOINTS	AWS STS Regionale Endpunkte
AWS_USE_DUALSTACK_ENDPOINT	Dual-Stack- und FIPS-Endpunkte
AWS_USE_FIPS_ENDPOINT	Dual-Stack- und FIPS-Endpunkte
AWS_WEB_IDENTITY_TOKEN_FILE	Nehmen Sie die Rolle des Anbieters von Anmeldeinformationen an

Liste der JVM-Systemeigenschaften

Sie können die folgenden JVM-Systemeigenschaften für die AWS SDK für Java und die AWS SDK für Kotlin (als Ziel für die JVM) verwenden. Anweisungen the section called “Wie legt man die JVM-Systemeigenschaften fest” zum Einstellen der JVM-Systemeigenschaften finden Sie unter.

Einstellungsname	Details
`aws.accessKeyId`	AWS Zugriffstasten
`aws.accountId`	Kontobasierte Endpunkte
`aws.accountIdEndpointMode`	Kontobasierte Endpunkte
`aws.authSchemePreference`	Authentifizierungsschema
`aws.configFile`	Suchen und Ändern des Speicherorts der geteilten `credentials` Dateien `config`AWS SDKs und Tools
`aws.defaultsMode`	Standardeinstellungen für die intelligente Konfiguration
`aws.disableEc2MetadataV1`	Anbieter von IMDS-Anmeldeinformationen
`aws.disableHostPrefixInjection`	Host-Präfix-Injektion
`aws.disableRequestCompression`	Komprimierung anfordern
`aws.disableS3ExpressAuth`	S3 Express One Zone-Sitzungsauthentifizierung

Einstellungsname	Details	
`aws.ec2MetadataServiceEndpoint`	Anbieter von IMDS-Anmeldeinformationen	
`aws.ec2MetadataServiceEndpointMode`	Anbieter von IMDS-Anmeldeinformationen	
`aws.endpointDiscoveryEnabled`	Erkennung von Endpunkten	
`aws.endpointUrl`	Servicespezifische Endpunkte	
`aws.endpointUrl<ServiceName>`	Servicespezifische Endpunkte	
`aws.ignoreConfiguredEndpointUrls`	Servicespezifische Endpunkte	
`aws.maxAttempts`	Verhalten wiederholen	
`aws.profile`	Verwenden von geteilten `credentials` Dateien `config` und Dateien zur globalen Konfiguration von AWS SDKs Tools	
`aws.region`	AWS-Region	
`aws.requestChecksumCalculation`	Schutz der Datenintegrität für Amazon S3	

Einstellungsname	Details
`aws.requestMinCompressionSizeBytes`	Komprimierung anfordern
`aws.responseChecksumValidation`	Schutz der Datenintegrität für Amazon S3
`aws.retryMode`	Verhalten wiederholen
`aws.roleArn`	Nehmen Sie die Rolle des Anbieters von Anmeldeinformationen an
`aws.roleSessionName`	Nehmen Sie die Rolle des Anbieters von Anmeldeinformationen an
`aws.s3DisableMultiRegionAccessPoints`	Mulitiregionale Amazon-S3-Zugriffspunkte
`aws.s3UseArnRegion`	Amazon-S3-Zugriffspunkte
`aws.secretAccessKey`	AWS Zugriffstasten
`aws.sessionToken`	AWS Zugriffstasten
`aws.sharedCredentialsFile`	Suchen und Ändern des Speicherorts der geteilten `credentials` Dateien `config`AWS SDKs und Tools

Einstellungsname	Details
`aws.useDualstackEndpoint`	Dual-Stack- und FIPS-Endpunkte
`aws.useFipsEndpoint`	Dual-Stack- und FIPS-Endpunkte
`aws.webIdentityTokenFile`	Nehmen Sie die Rolle des Anbieters von Anmeldeinformationen an
`sdk.ua.appId`	Application ID

AWS SDKs und Tools standardisierte Anbieter von Anmeldeinformationen

Viele Anbieter von Anmeldeinformationen wurden auf einheitliche Standardwerte standardisiert und funktionieren in vielen SDKs auf die gleiche Weise. Diese Konsistenz erhöht die Produktivität und Klarheit bei der Codierung mehrerer SDKs. Alle Einstellungen können im Code überschrieben werden. Einzelheiten finden Sie in Ihrer spezifischen SDK-API.

> **Important**
>
> Nicht alle SDKs unterstützen alle Anbieter oder sogar alle Aspekte innerhalb eines Anbieters.

Themen

- Machen Sie sich mit der Kette der Anbieter von Anmeldeinformationen vertraut
- SDK-specific und toolspezifische Anbieterketten für Anmeldeinformationen
- AWS Zugriffstasten
- Anbieter von Anmeldedaten
- Übernehmen Sie die Rolle Credential Provider
- Anbieter von Container-Anmeldeinformationen

- IAM Identity Center-Anmeldeinformationsanbieter
- IMDS-Anmeldeinformationsanbieter
- Anbieter von Prozessanmeldeinformationen

Machen Sie sich mit der Kette der Anbieter von Anmeldeinformationen vertraut

Alle SDKs haben eine Reihe von Stellen (oder Quellen), an denen sie nach gültigen Anmeldeinformationen suchen, die sie verwenden können, um eine Anfrage an einen zu stellen. AWS-Service Nachdem gültige Anmeldeinformationen gefunden wurden, wird die Suche beendet. Diese systematische Suche wird als Credential Provider Chain bezeichnet.

Wenn Sie einen der standardisierten Anbieter für Anmeldeinformationen verwenden, versuchen die AWS SDKs immer, Anmeldeinformationen automatisch zu erneuern, wenn sie ablaufen. Die integrierte Kette von Anbietern von Anmeldeinformationen bietet Ihrer Anwendung die Möglichkeit, Ihre Anmeldeinformationen zu aktualisieren, unabhängig davon, welchen Anbieter Sie in der Kette verwenden. Für das SDK ist dafür kein zusätzlicher Code erforderlich.

Obwohl die einzelnen SDKs unterschiedlich sind, enthalten sie in den meisten Fällen Quellen wie die folgenden:

Anbieter von Anmeldeinformationen	Description
AWS Zugriffstasten	AWS Zugriffsschlüssel für einen IAM-Benutzer (wie`AWS_ACCESS_KEY_ID` , und`AWS_SECRET_ACCESS_KEY`).
Verbunden mit Web-Identität oder OpenID Connect— Nehmen Sie die Rolle des Anbieters von Anmeldeinformationen an	Melden Sie sich mit einem bekannten externen Identitätsanbieter (IdP) an, z. B. Login with Amazon, Facebook, Google oder einem anderen OpenID Connect (OIDC) -kompatiblen IdP. Nehmen Sie die Berechtigungen einer IAM-Rolle an, indem Sie ein JSON-Webtoken (JWT) von () verwenden. AWS -Security-Token-Service AWS STS
Anbieter von Anmeldedaten	Rufen Sie Anmeldeinformationen für eine neue oder bestehende Konsolensitzung ab, bei der Sie angemeldet sind.

Anbieter von Anmeldeinformationen	Description
IAM Identity Center-Anmeldeinformationsanbieter	Holen Sie sich Anmeldeinformationen von AWS IAM Identity Center.
Übernehmen Sie die Rolle Credential Provider	Erhalten Sie Zugriff auf andere Ressourcen, indem Sie die Berechtigungen einer IAM-Rolle übernehmen. (Rufen Sie temporäre Anmeldeinformationen für eine Rolle ab und verwenden Sie sie anschließend).
Anbieter von Container-Anmeldeinformationen	Anmeldeinformationen für Amazon Elastic Container Service (Amazon ECS) und Amazon Elastic Kubernetes Service (Amazon EKS). Der Anbieter für Container-Anmeldeinformationen ruft Anmeldeinformationen für die containerisierte Anwendung des Kunden ab.
Anbieter von Prozessanmeldeinformationen	Benutzerdefinierter Anbieter für Anmeldeinformationen. Rufen Sie Ihre Anmeldeinformationen aus einer externen Quelle oder einem externen Prozess ab, einschließlich IAM Roles Anywhere.
IMDS-Anmeldeinformationsanbieter	Anmeldeinformationen für das Amazon Elastic Compute Cloud (Amazon EC2) -Instanzprofil. Ordnen Sie jeder Ihrer EC2-Instances eine IAM-Rolle zu. Temporäre Anmeldeinformationen für diese Rolle werden dem Code zur Verfügung gestellt, der in der Instance ausgeführt wird. Die Anmeldeinformationen werden über den Amazon-EC2-Metadaten-Service bereitgestellt.

Für jeden Schritt in der Kette gibt es mehrere Möglichkeiten, Einstellungswerte zuzuweisen. Einstellungswerte, die im Code angegeben sind, haben immer Vorrang. Es gibt jedoch auch Umgebungsvariablen und dieVerwenden von geteilten `credentials` Dateien `config` und Dateien zur globalen Konfiguration AWS SDKs und Tools. Weitere Informationen finden Sie unter Vorrang der Einstellungen.

SDK-specific und toolspezifische Anbieterketten für Anmeldeinformationen

Um direkt zu den spezifischen Details der Anmeldeinformationsanbieterkette Ihres SDK oder Tools zu gelangen, wählen Sie Ihr SDK oder Tool aus den folgenden Optionen aus:

- AWS CLI
- SDK for C++
- SDK for Go
- SDK für Java
- SDK für JavaScript
- SDK für Kotlin
- SDK for .NET
- SDK for PHP
- SDK for Python (Boto3)
- SDK for Ruby
- SDK für Rust
- SDK für Swift
- Tools für PowerShell

AWS Zugriffstasten

> **Warning**
>
> Um Sicherheitsrisiken zu vermeiden, sollten Sie IAM-Benutzer nicht zur Authentifizierung verwenden, wenn Sie speziell entwickelte Software entwickeln oder mit echten Daten arbeiten. Verwenden Sie stattdessen den Verbund mit einem Identitätsanbieter wie AWS IAM Identity Center.

AWS Zugriffsschlüssel für einen IAM-Benutzer können als Ihre AWS Anmeldeinformationen verwendet werden. Das AWS SDK verwendet diese AWS Anmeldeinformationen automatisch, um API-Anfragen zu signieren AWS, sodass Ihre Workloads sicher und bequem auf Ihre AWS Ressourcen und Daten zugreifen können. Es wird empfohlen, immer die zu verwenden,

`aws_session_token` damit die Anmeldeinformationen temporär sind und nach Ablauf nicht mehr gültig sind. Die Verwendung langfristiger Anmeldeinformationen wird nicht empfohlen.

> **Note**
>
> Wenn AWS diese temporären Anmeldeinformationen nicht aktualisiert werden AWS können, kann dies die Gültigkeit der Anmeldeinformationen verlängern, sodass Ihre Workloads nicht beeinträchtigt werden.

Die gemeinsam genutzte AWS `credentials` Datei ist der empfohlene Speicherort für Anmeldeinformationen, da sie sich sicher außerhalb der Quellverzeichnisse der Anwendung befindet und von den SDK-specific Einstellungen der gemeinsam genutzten `config` Datei getrennt ist.

Weitere Informationen zu AWS Anmeldeinformationen und zur Verwendung von Zugriffsschlüsseln finden Sie unter AWS Sicherheitsanmeldeinformationen und Verwaltung von Zugriffsschlüsseln für IAM-Benutzer im IAM-Benutzerhandbuch.

Konfigurieren Sie diese Funktionalität wie folgt:

`aws_access_key_id`- Einstellung für gemeinsam genutzte AWS **`config`** Dateien, **`aws_access_key_id`**- Einstellung für gemeinsam genutzte AWS **`credentials`** Dateien (empfohlene Methode), **`AWS_ACCESS_KEY_ID`**- Umgebungsvariable, **`aws.accessKeyId`**- JVM-Systemeigenschaft: nur Java/Kotlin

Gibt den AWS Zugriffsschlüssel an, der als Teil der Anmeldeinformationen zur Authentifizierung des Benutzers verwendet wird.

`aws_secret_access_key`- Einstellung für gemeinsam genutzte AWS **`config`** Dateien, **`aws_secret_access_key`**- Einstellung für gemeinsam genutzte AWS **`credentials`** Dateien (empfohlene Methode), **`AWS_SECRET_ACCESS_KEY`**- Umgebungsvariable, **`aws.secretAccessKey`**- JVM-Systemeigenschaft: nur Java/Kotlin

Gibt den AWS geheimen Schlüssel an, der als Teil der Anmeldeinformationen zur Authentifizierung des Benutzers verwendet wird.

`aws_session_token`- Einstellung für gemeinsam genutzte AWS **`config`** Dateien, **`aws_session_token`**- Einstellung für gemeinsam genutzte AWS **`credentials`** Dateien (empfohlene Methode), **`AWS_SESSION_TOKEN`**- Umgebungsvariable, **`aws.sessionToken`**- JVM-Systemeigenschaft: nur Java/Kotlin

> Gibt ein AWS Sitzungstoken an, das als Teil der Anmeldeinformationen zur Authentifizierung des Benutzers verwendet wird. Sie erhalten diesen Wert als Teil der temporären Anmeldeinformationen, die bei erfolgreichen Anfragen zur Übernahme einer Rolle zurückgegeben werden. Ein Sitzungs-Token ist nur erforderlich, wenn Sie manuell temporäre Anmeldeinformationen angeben. Wir empfehlen jedoch, immer temporäre Sicherheitsanmeldedaten statt langfristiger Anmeldeinformationen zu verwenden. Sicherheitsempfehlungen finden Sie unter Bewährte Sicherheitsmethoden in IAM.

Anweisungen zum Abrufen dieser Werte finden Sie unterVerwendung von kurzfristigen Anmeldeinformationen zur Authentifizierung AWS SDKs und Tools.

Beispiel für das Einstellen dieser erforderlichen Werte in der `config` `credentials` OR-Datei:

```
[default]
aws_access_key_id = AKIAIOSFODNN7EXAMPLE
aws_secret_access_key = wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
aws_session_token = AQoEXAMPLEH4aoAH0gNCAPy...truncated...zrkuWJOgQs8IZZaIv2BXIa2R4Olgk
```

Linux/macOS Beispiel für das Setzen von Umgebungsvariablen über die Befehlszeile:

```
export AWS_ACCESS_KEY_ID=AKIAIOSFODNN7EXAMPLE
export AWS_SECRET_ACCESS_KEY=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
export
 AWS_SESSION_TOKEN=AQoEXAMPLEH4aoAH0gNCAPy...truncated...zrkuWJOgQs8IZZaIv2BXIa2R4Olgk
```

Windows-Beispiel für das Setzen von Umgebungsvariablen über die Befehlszeile:

```
setx AWS_ACCESS_KEY_ID AKIAIOSFODNN7EXAMPLE
setx AWS_SECRET_ACCESS_KEY wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
setx
 AWS_SESSION_TOKEN AQoEXAMPLEH4aoAH0gNCAPy...truncated...zrkuWJOgQs8IZZaIv2BXIa2R4Olgk
```

Support durch AWS SDKs und Tools

Die folgenden SDKs unterstützen die in diesem Thema beschriebenen Funktionen und Einstellungen. Alle teilweisen Ausnahmen werden vermerkt. Alle Einstellungen für JVM-Systemeigenschaften werden AWS SDK für Kotlin nur von AWS SDK für Java und vom unterstützt.

SDK	Unterstützt	Hinweise oder weitere Informationen
AWS CLI v2	Ja	
SDK for C++	Ja	Die gemeinsam genutzte `config` Datei wird nicht unterstützt.
SDK for Go V2 (1.x)	Ja	
SDK for Go 1.x (V1)	Ja	Um die Einstellungen für gemeinsam genutzte `config` Dateien zu verwenden, müssen Sie das Laden aus der Konfigurationsdatei aktivieren. Weitere Informationen finden Sie unter Sessions.
SDK for Java 2.x	Ja	
SDK for Java 1.x	Ja	
SDK für 3.x JavaScript	Ja	
SDK für 2.x JavaScript	Ja	
SDK für Kotlin	Ja	
SDK for .NET 4.x	Ja	
SDK for .NET 3.x	Ja	
SDK for PHP 3.x	Ja	
SDK for Python (Boto3)	Ja	
SDK for Ruby 3.x	Ja	
SDK für Rust	Ja	

SDK	Unterstützt	Hinweise oder weitere Informationen
SDK für Swift	Ja	
Tools für PowerShell V5	Ja	
Tools für V4 PowerShell	Ja	Umgebungsvariablen werden nicht unterstützt.

Anbieter von Anmeldedaten

Sie können Ihre vorhandenen Anmeldedaten für die AWS Management Console verwenden, um kurzfristige Anmeldeinformationen zu erwerben, die für den programmatischen Zugriff verwendet werden können. Nachdem Sie den browserbasierten Authentifizierungsablauf abgeschlossen haben, AWS generiert es temporäre Anmeldeinformationen, die in allen lokalen Entwicklungstools wie AWS CLI, AWS Tools for PowerShell und AWS SDKs funktionieren.

Um diese Anmeldeinformationen zu generieren, führen Sie den `aws login` Befehl in der AWS CLI oder das `Invoke-AWSLogin` Cmdlet in AWS Tools for aus. PowerShell Die resultierenden kurzfristigen Anmeldeinformationen werden lokal zwischengespeichert, wo sie von den SDKs wiederverwendet werden können. AWS Die kurzfristigen Anmeldeinformationen laufen in 15 Minuten ab, aber die CLI und die SDKs aktualisieren sie bei Bedarf automatisch bis zu 12 Stunden. Wenn das Aktualisierungstoken abläuft, werden Sie aufgefordert, sich erneut über die CLI oder anzumelden PowerShell.

Mit dem Login-Befehl wird das von Ihnen angegebene Profil mit der `login_session` Einstellung aktualisiert, in der die Identität der Verwaltungskonsolensitzung gespeichert wird, die Sie während des Anmelde-Workflows ausgewählt haben.

```
[profile console]
login_session = arn:aws:iam::0123456789012:user/username
region = us-west-2
```

Standardmäßig werden die kurzfristigen Anmeldeinformationen und das Aktualisierungstoken in einer JSON-Datei im `~/.aws/login/cache` Verzeichnis unter Linux und macOS oder `%USERPROFILE%\.aws\login\cache` unter Windows gespeichert. Der Dateiname basiert auf dem Namen der Anmeldesitzung. Sie können das Verzeichnis überschreiben, indem Sie die `AWS_LOGIN_CACHE_DIRECTORY` Umgebungsvariable setzen.

Einstellungen des Anmeldeanbieters

Konfigurieren Sie diese Funktionalität wie folgt:

`AWS_LOGIN_CACHE_DIRECTORY`- Umgebungsvariable

Alternatives Verzeichnis, in dem die CLI und die SDKs die zwischengespeicherten Anmeldeinformationen speichern, die einem Anmeldesitzungsprofil zugeordnet sind.

Standardwert: `~/.aws/login/cache` unter Linux und macOS oder `%USERPROFILE%\.aws\login\cache` unter Windows.

Support durch AWS SDKs und Tools

Die folgenden SDKs unterstützen die in diesem Thema beschriebenen Funktionen und Einstellungen. Alle teilweisen Ausnahmen werden vermerkt. Alle Einstellungen für JVM-Systemeigenschaften werden AWS SDK für Kotlin nur von AWS SDK für Java und vom unterstützt.

SDK	Unterstützt	Hinweise oder weitere Informationen
AWS CLI v2	Ja	
SDK for C++	Ja	
SDK for Go V2 (1.x)	Nein	
SDK for Go 1.x (V1)	Ja	
SDK for Java 2.x	Ja	
SDK for Java 1.x	Nein	
SDK für 3.x JavaScript	Ja	
SDK für 2.x JavaScript	Nein	
SDK für Kotlin	Ja	
SDK for .NET 4.x	Ja	

SDK	Unterstützt	Hinweise oder weitere Informationen
SDK for .NET 3.x	Ja	
SDK for PHP 3.x	Ja	
SDK for Python (Boto3)	Ja	Erfordert CRT
SDK for Ruby 3.x	Ja	
SDK für Rust	Ja	
Tools für PowerShell V5	Ja	
Tools für V4 PowerShell	Nein	

Übernehmen Sie die Rolle Credential Provider

> **Note**
>
> Hilfe zum Verständnis des Layouts von Einstellungsseiten oder zur Interpretation der nachfolgenden Tabelle Support by AWS SDKs and Tools finden Sie unterDie Einstellungsseiten dieses Handbuchs verstehen.

Die Übernahme einer Rolle beinhaltet die Verwendung einer Reihe temporärer Sicherheitsanmeldedaten für den Zugriff auf AWS Ressourcen, auf die Sie sonst möglicherweise keinen Zugriff hätten. Diese temporären Anmeldeinformationen bestehen aus einer Zugriffsschlüssel-ID, einem geheimen Zugriffsschlüssel und einem Sicherheits-Token.

Um Ihr SDK oder Tool so einzurichten, dass es eine Rolle übernimmt, müssen Sie zunächst eine bestimmte Rolle erstellen oder identifizieren, die Sie übernehmen möchten. IAM-Rollen werden durch eine Rolle mit dem Amazon Resource Name (ARN) eindeutig identifiziert. Rollen bauen Vertrauensbeziehungen zu einer anderen Entität auf. Die vertrauenswürdige Entität, die die Rolle verwendet, kann ein anderer AWS-Service AWS-Konto, ein Web-Identitätsanbieter oder ein OIDC- oder SAML-Verbund sein.

Nachdem die IAM-Rolle identifiziert wurde und diese Rolle Ihnen vertraut, können Sie Ihr SDK oder Tool so konfigurieren, dass die von der Rolle gewährten Berechtigungen verwendet werden. Verwenden Sie dazu die folgenden Einstellungen.

Anleitungen zu den ersten Schritten mit diesen Einstellungen finden Sie Übernahme einer Rolle mit AWS Anmeldeinformationen zur Authentifizierung AWS SDKs und Tools in diesem Handbuch.

Nehmen Sie die Einstellungen des Anbieters für Anmeldeinformationen an

Konfigurieren Sie diese Funktionalität wie folgt:

`credential_source`- Einstellung für gemeinsam genutzte AWS **`config`** Dateien

Wird innerhalb von Amazon EC2 EC2-Instances oder Amazon Elastic Container Service-Containern verwendet, um anzugeben, wo das SDK oder Tool Anmeldeinformationen finden kann, die berechtigt sind, die Rolle anzunehmen, die Sie mit dem `role_arn` Parameter angeben.

Standardwert: Keiner

Zulässige Werte:

- Umgebung — Gibt an, dass das SDK oder Tool Quellanmeldedaten aus den Umgebungsvariablen `AWS_ACCESS_KEY_IDund` abrufen sollAWS_SECRET_ACCESS_KEY.
- Ec2InstanceMetadata— Gibt an, dass das SDK oder Tool die dem EC2-Instance-Profil zugeordnete IAM-Rolle verwenden soll, um Quellanmeldedaten abzurufen.
- EcsContainer— Gibt an, dass das SDK oder Tool die dem Amazon ECS-Container zugeordnete IAM-Rolle oder die dem Amazon EKS-Container zugeordnete IAM-Rolle verwenden soll, um Quellanmeldedaten abzurufen.

Sie können `credential_source` und `source_profile` nicht im selben Profil angeben.

Beispiel für die Einstellung in einer `config` Datei, um anzugeben, dass Anmeldeinformationen von Amazon EC2 bezogen werden sollen:

```
credential_source = Ec2InstanceMetadata
role_arn = arn:aws:iam::123456789012:role/my-role-name
```

`duration_seconds`- Einstellung für gemeinsam genutzte AWS **`config`** Dateien

Gibt die maximale Dauer der Rollensitzung in Sekunden an.

Diese Einstellung gilt nur, wenn das Profil angibt, dass eine Rolle übernommen werden soll.

Standardwert: 3600 Sekunden (eine Stunde)

Gültige Werte: Der Wert kann zwischen 900 Sekunden (15 Minuten) und der für die Rolle konfigurierten Einstellung für die maximale Sitzungsdauer liegen (die maximal 43200 Sekunden oder 12 Stunden betragen kann). Weitere Informationen finden Sie im IAM-Benutzerhandbuch unter Einstellung „Maximale Sitzungsdauer“ für eine Rolle anzeigen.

Beispiel für die Einstellung dieser Einstellung in einer `config` Datei:

```
duration_seconds = 43200
```

`external_id`- Einstellung für gemeinsam genutzte AWS **`config`** Dateien

Gibt eine eindeutige Kennung an, die von Dritten verwendet wird, um eine Rolle in den Konten ihrer Kunden zu übernehmen.

Diese Einstellung gilt nur, wenn das Profil angibt, dass eine Rolle übernommen werden soll und die Vertrauensrichtlinie für die Rolle einen Wert für erfordert`ExternalId`. Der Wert ist dem `ExternalId` Parameter zugeordnet, der an den `AssumeRole` Vorgang übergeben wird, wenn das Profil eine Rolle angibt.

Standardwert: Keiner.

Gültige Werte: Weitere Informationen finden Sie im IAM-Benutzerhandbuch unter So verwenden Sie eine externe ID, wenn Sie Dritten Zugriff auf Ihre AWS Ressourcen gewähren.

Beispiel für die Einstellung in einer `config` Datei:

```
external_id = unique_value_assigned_by_3rd_party
```

`mfa_serial`- Einstellung für gemeinsam genutzte AWS **`config`** Dateien

Gibt die Identifikations- oder Seriennummer eines Geräts mit Multi-Faktor-Authentifizierung (MFA) an, das der Benutzer verwenden muss, wenn er eine Rolle übernimmt.

Erforderlich, wenn Sie eine Rolle übernehmen, bei der die Vertrauensrichtlinie für diese Rolle eine Bedingung beinhaltet, die eine MFA-Authentifizierung erfordert. Weitere Informationen zu MFA finden Sie unter AWS Multi-factor Authentifizierung in IAM im IAM-Benutzerhandbuch.

Standardwert: Keiner.

Gültige Werte: Der Wert kann entweder eine Seriennummer für ein Hardwaregerät (z. B.GAHT12345678) oder ein Amazon-Ressourcenname (ARN) für ein virtuelles MFA-Gerät sein. Das Format des ARN ist: `arn:aws:iam::`*`account-id`*`:mfa/`*`mfa-device-name`*

Beispiel für die Einstellung in einer `config` Datei:

In diesem Beispiel wird davon ausgegangen, dass ein virtuelles MFA-Gerät namens`MyMFADevice`, für das Konto erstellt und für einen Benutzer aktiviert wurde.

```
mfa_serial = arn:aws:iam::123456789012:mfa/MyMFADevice
```

`role_arn`- Einstellung für gemeinsam genutzte AWS **`config`** Dateien, **`AWS_ROLE_ARN`**-Umgebungsvariable, **`aws.roleArn`**- JVM-Systemeigenschaft: nur Java/Kotlin

Gibt den Amazon-Ressourcennamen (ARN) einer IAM-Rolle an, die Sie verwenden möchten, um mit diesem Profil angeforderte Operationen auszuführen.

Standardwert: Keiner.

Gültige Werte: Der Wert muss der ARN einer IAM-Rolle sein, der wie folgt formatiert ist: `arn:aws:iam::`*`account-id`*`:role/`*`role-name`*

Darüber hinaus müssen Sie auch eine der folgenden Einstellungen angeben:

- `source_profile`— Um ein anderes Profil zu identifizieren, das verwendet werden soll, um Anmeldeinformationen zu finden, die berechtigt sind, die Rolle in diesem Profil zu übernehmen.
- `credential_source`— Um entweder Anmeldeinformationen zu verwenden, die durch die aktuellen Umgebungsvariablen identifiziert wurden, oder Anmeldeinformationen, die einem Amazon EC2 EC2-Instance-Profil zugeordnet sind, oder einer Amazon ECS-Container-Instance.
- `web_identity_token_file`— Um öffentliche Identitätsanbieter oder einen OpenID Connect (OIDC) -kompatiblen Identitätsanbieter für Benutzer zu verwenden, die in einer Mobil- oder Webanwendung authentifiziert wurden.

`role_session_name`- Einstellung für gemeinsam genutzte Dateien AWS **`config`**, **`AWS_ROLE_SESSION_NAME`**- Umgebungsvariable, **`aws.roleSessionName`**- JVM-Systemeigenschaft: nur Java/Kotlin

Gibt den Namen an, der der Rollensitzung zugeordnet werden soll. Dieser Name erscheint in den AWS CloudTrail Protokollen für Einträge, die mit dieser Sitzung verknüpft sind, was bei der

Prüfung nützlich sein kann. Einzelheiten finden Sie unter CloudTrailUserIdentity-Element im AWS CloudTrail Benutzerhandbuch.

Standardwert: Ein optionaler Parameter. Wenn Sie diesen Wert nicht angeben, wird automatisch ein Sitzungsname generiert, wenn das Profil eine Rolle annimmt.

Gültige Werte: Werden für den `RoleSessionName` Parameter bereitgestellt, wenn die AWS API AWS CLI oder die `AssumeRole` Operation (oder Operationen wie die `AssumeRoleWithWebIdentity` Operation) in Ihrem Namen aufruft. Der Wert wird Teil des angenommenen Rollenbenutzers Amazon Resource Name (ARN), den Sie abfragen können, und wird als Teil der CloudTrail Protokolleinträge für Operationen angezeigt, die von diesem Profil aufgerufen werden.

`arn:aws:sts::`*`123456789012`*`:assumed-role/`*`my-role-name`*`/`***`my-role_session_name`***.

Beispiel für die Einstellung in einer `config` Datei:

```
role_session_name = my-role-session-name
```

`source_profile`- Einstellung für gemeinsam genutzte AWS **`config`** Dateien

Gibt ein anderes Profil an, dessen Anmeldeinformationen verwendet werden, um die Rolle anzunehmen, die in der `role_arn` Einstellung im ursprünglichen Profil angegeben ist. Informationen zur Verwendung von Profilen in geteilten `credentials` Dateien AWS `config` und Dateien finden Sie unterGeteilte `credentials` Dateien `config` und Dateien.

Wenn Sie ein Profil angeben, bei dem es sich auch um ein Rollenübernahmeprofil handelt, wird jede Rolle der Reihe nach übernommen, um die Anmeldeinformationen vollständig aufzulösen. Diese Kette wird unterbrochen, wenn das SDK auf ein Profil mit Anmeldeinformationen trifft. Die Rollenverkettung begrenzt Ihre Rollensitzung AWS CLI oder Ihre AWS API-Rollensitzung auf maximal eine Stunde und kann nicht verlängert werden. Weitere Informationen finden Sie im IAM-Benutzerhandbuch unter Begriffe und Konzepte für Rollen.

Standardwert: Keiner.

Gültige Werte: Eine Textzeichenfolge, die aus dem Namen eines Profils besteht, das in den `credentials` Dateien `config` und definiert ist. Sie müssen auch einen Wert für `role_arn` im aktuellen Profil angeben.

Sie können `credential_source` und `source_profile` nicht im selben Profil angeben.

Beispiel für die Einstellung in einer Konfigurationsdatei:

```
[profile A]
source_profile = B
role_arn =  arn:aws:iam::123456789012:role/RoleA
role_session_name = ProfileARoleSession

[profile B]
credential_process = ./aws_signing_helper credential-process --certificate /
path/to/certificate --private-key /path/to/private-key --trust-anchor-
arn arn:aws:rolesanywhere:region:account:trust-anchor/TA_ID --profile-
arn arn:aws:rolesanywhere:region:account:profile/PROFILE_ID --role-arn
 arn:aws:iam::account:role/ROLE_ID
```

Im vorherigen Beispiel weist das A Profil das SDK oder Tool an, automatisch nach den Anmeldeinformationen für das verknüpfte B Profil zu suchen. In diesem Fall verwendet das B Profil das Credential Helper-Tool, das von bereitgestellt wirdAuthentifizierung AWS SDKs und Tools mit IAM Roles Anywhere, um Anmeldeinformationen für das AWS SDK abzurufen. Diese temporären Anmeldeinformationen werden dann von Ihrem Code für den Zugriff auf AWS Ressourcen verwendet. An die angegebene Rolle müssen IAM-Berechtigungsrichtlinien angehängt sein, die die Ausführung des angeforderten Codes ermöglichen, z. B. der Befehl oder die API-Methode. AWS-Service Für jede Aktion, die vom Profil A ausgeführt wird, ist der Name der Rollensitzung in den CloudTrail Protokollen enthalten.

Als zweites Beispiel für Rollenverkettung kann die folgende Konfiguration verwendet werden, wenn Sie eine Anwendung auf einer Amazon Elastic Compute Cloud-Instance haben und möchten, dass diese Anwendung eine andere Rolle übernimmt.

```
[profile A]
source_profile = B
role_arn =  arn:aws:iam::123456789012:role/RoleA
role_session_name = ProfileARoleSession

[profile B]
credential_source=Ec2InstanceMetadata
```

Profile verwendet A die Anmeldeinformationen der Amazon EC2 EC2-Instance, um die angegebene Rolle anzunehmen, und erneuert die Anmeldeinformationen automatisch.

`web_identity_token_file`- Einstellung für gemeinsam genutzte AWS **`config`** Dateien, **`AWS_WEB_IDENTITY_TOKEN_FILE`**- Umgebungsvariable, **`aws.webIdentityTokenFile`**- JVM-Systemeigenschaft: nur Java/Kotlin

Gibt den Pfad zu einer Datei an, die ein Zugriffstoken von einem unterstützten OAuth 2.0-Anbieter oder OpenID Connect ID-Identitätsanbieter enthält.

Diese Einstellung ermöglicht die Authentifizierung mithilfe von Web Identity Federation-Anbietern wie Google, Facebook und Amazon und vielen anderen. Das SDK oder das Entwicklertool lädt den Inhalt dieser Datei und übergibt ihn als `WebIdentityToken` Argument, wenn es den `AssumeRoleWithWebIdentity` Vorgang in Ihrem Namen aufruft.

Standardwert: Keiner.

Gültige Werte: Dieser Wert muss ein Pfad und ein Dateiname sein. Die Datei muss ein OAuth 2.0-Zugriffstoken oder ein OpenID Connect-Token enthalten, das Ihnen von einem Identitätsanbieter zur Verfügung gestellt wurde. Relative Pfade werden als relativ zum Arbeitsverzeichnis des Prozesses behandelt.

Support durch AWS SDKs und Tools

Die folgenden SDKs unterstützen die in diesem Thema beschriebenen Funktionen und Einstellungen. Alle teilweisen Ausnahmen werden vermerkt. Alle Einstellungen für JVM-Systemeigenschaften werden AWS SDK für Kotlin nur von AWS SDK für Java und vom unterstützt.

SDK	Unterstützt	Hinweise oder weitere Informationen
AWS CLI v2	Ja	
SDK for C++	Teilweise	`credential_source` wird nicht unterstützt. `duration_seconds` nicht unterstützt. `mfa_serial` nicht unterstützt.
SDK for Go V2 (1.x)	Ja	
SDK for Go 1.x (V1)	Ja	Um die Einstellungen für gemeinsam genutzte `config` Dateien zu verwenden, müssen Sie das Laden aus der Konfigurationsdatei aktivieren. Weitere Informationen finden Sie unter Sessions.

SDK	Unterstützt	Hinweise oder weitere Informationen
SDK for Java 2.x	Teilweise	`mfa_serial` wird nicht unterstützt. `duration_seconds` nicht unterstützt.
SDK for Java 1.x	Teilweise	`credential_source` wird nicht unterstützt. `mfa_serial` nicht unterstützt. JVM-Systemeigenschaften werden nicht unterstützt.
SDK für 3.x JavaScript	Ja	
SDK für 2.x JavaScript	Teilweise	`credential_source` wird nicht unterstützt.
SDK für Kotlin	Ja	
SDK for .NET 4.x	Ja	
SDK for .NET 3.x	Ja	
SDK for PHP 3.x	Ja	
SDK for Python (Boto3)	Ja	
SDK for Ruby 3.x	Ja	
SDK für Rust	Ja	
SDK für Swift	Ja	
Tools für PowerShell V5	Ja	
Tools für V4 PowerShell	Ja	

Anbieter von Container-Anmeldeinformationen

> **Note**
>
> Hilfe zum Verständnis des Layouts von Einstellungsseiten oder zur Interpretation der nachfolgenden Tabelle Support by AWS SDKs and Tools finden Sie unterDie Einstellungsseiten dieses Handbuchs verstehen.

Der Anbieter für Container-Anmeldeinformationen ruft Anmeldeinformationen für die containerisierte Anwendung des Kunden ab. Dieser Anmeldeinformationsanbieter ist für Kunden von Amazon Elastic Container Service (Amazon ECS) und Amazon Elastic Kubernetes Service (Amazon EKS) nützlich. SDKs versuchen, Anmeldeinformationen über eine GET-Anfrage vom angegebenen HTTP-Endpunkt zu laden.

Wenn Sie Amazon ECS verwenden, empfehlen wir Ihnen, eine Task-IAM-Rolle zu verwenden, um die Isolierung, Autorisierung und Überprüfbarkeit von Anmeldeinformationen zu verbessern. Nach der Konfiguration legt Amazon ECS die `AWS_CONTAINER_CREDENTIALS_RELATIVE_URI` Umgebungsvariable fest, die die SDKs und Tools zum Abrufen von Anmeldeinformationen verwenden. Informationen zur Konfiguration von Amazon ECS für diese Funktionalität finden Sie unter Task IAM-Rolle im Amazon Elastic Container Service Developer Guide.

Wenn Sie Amazon EKS verwenden, empfehlen wir Ihnen, Amazon EKS Pod Identity zu verwenden, um die Isolierung von Anmeldeinformationen, die geringsten Rechte, die Überprüfbarkeit, den unabhängigen Betrieb, die Wiederverwendbarkeit und die Skalierbarkeit zu verbessern. Sowohl Ihre Pod- als auch eine IAM-Rolle sind mit einem Kubernetes-Servicekonto verknüpft, um die Anmeldeinformationen für Ihre Anwendungen zu verwalten. Weitere Informationen zu Amazon EKS Pod Identity finden Sie unter Amazon EKS Pod Identities im Amazon EKS-Benutzerhandbuch. Nach der Konfiguration legt Amazon EKS die Umgebungsvariablen `AWS_CONTAINER_CREDENTIALS_FULL_URI` und die `AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE` Umgebungsvariablen fest, die die SDKs und Tools zum Abrufen von Anmeldeinformationen verwenden. Informationen zur Einrichtung finden Sie unter Einrichten des Amazon EKS Pod Identity Agent im Amazon EKS-Benutzerhandbuch oder Amazon EKS Pod Identity vereinfacht IAM-Berechtigungen für Anwendungen auf Amazon EKS-Clustern auf der AWS Blog-Website.

Konfigurieren Sie diese Funktionalität wie folgt:

`AWS_CONTAINER_CREDENTIALS_FULL_URI`- Umgebungsvariable

Gibt den vollständigen HTTP-URL-Endpunkt an, den das SDK bei der Anforderung von Anmeldeinformationen verwenden soll. Dies umfasst sowohl das Schema als auch den Host.

Standardwert: Keiner.

Gültige Werte: Gültiger URI.

Hinweis: Diese Einstellung ist eine Alternative zu *`AWS_CONTAINER_CREDENTIALS_RELATIVE_URI`* und wird nur verwendet, wenn sie nicht gesetzt *`AWS_CONTAINER_CREDENTIALS_RELATIVE_URI`* ist.

Linux/macOS Beispiel für das Setzen von Umgebungsvariablen über die Befehlszeile:

```
export AWS_CONTAINER_CREDENTIALS_FULL_URI=http://localhost/get-credentials
```

oder

```
export AWS_CONTAINER_CREDENTIALS_FULL_URI=http://localhost:8080/get-credentials
```

`AWS_CONTAINER_CREDENTIALS_RELATIVE_URI`- Umgebungsvariable

Gibt den relativen HTTP-URL-Endpunkt an, den das SDK bei der Anforderung von Anmeldeinformationen verwenden soll. Der Wert wird an den standardmäßigen Amazon ECS-Hostnamen von angehängt. `169.254.170.2`

Standardwert: Keiner.

Gültige Werte: Gültiger relativer URI.

Linux/macOS Beispiel für das Setzen von Umgebungsvariablen über die Befehlszeile:

```
export AWS_CONTAINER_CREDENTIALS_RELATIVE_URI=/get-credentials?a=1
```

`AWS_CONTAINER_AUTHORIZATION_TOKEN`- Umgebungsvariable

Gibt ein Autorisierungstoken im Klartext an. Wenn diese Variable gesetzt ist, legt das SDK den Authorization-Header in der HTTP-Anfrage mit dem Wert der Umgebungsvariablen fest.

Standardwert: Keiner.

Gültige Werte: Zeichenfolge.

Hinweis: Diese Einstellung ist eine Alternative zu *AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE* und wird nur verwendet, wenn sie nicht gesetzt *AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE* ist.

Linux/macOS Beispiel für das Setzen von Umgebungsvariablen über die Befehlszeile:

```
export AWS_CONTAINER_CREDENTIALS_FULL_URI=http://localhost/get-credential
export AWS_CONTAINER_AUTHORIZATION_TOKEN=Basic abcd
```

AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE- Umgebungsvariable

Gibt einen absoluten Dateipfad zu einer Datei an, die das Autorisierungstoken im Klartext enthält.

Standardwert: Keiner.

Gültige Werte: Zeichenfolge.

Linux/macOS Beispiel für das Setzen von Umgebungsvariablen über die Befehlszeile:

```
export AWS_CONTAINER_CREDENTIALS_FULL_URI=http://localhost/get-credential
export AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE=/path/to/token
```

Support durch AWS SDKs und Tools

Die folgenden SDKs unterstützen die in diesem Thema beschriebenen Funktionen und Einstellungen. Alle teilweisen Ausnahmen werden vermerkt. Alle Einstellungen für JVM-Systemeigenschaften werden AWS SDK für Kotlin nur von AWS SDK für Java und vom unterstützt.

SDK	Unterstützt	Hinweise oder weitere Informationen
AWS CLI v2	Ja	
SDK for C++	Ja	
SDK for Go V2 (1.x)	Ja	
SDK for Go 1.x (V1)	Ja	

SDK	Unterstützt	Hinweise oder weitere Informationen
SDK for Java 2.x	Ja	Wenn Lambda aktiviert SnapStart ist `AWS_CONTAINER_CREDENTIALS_FULL_URI` und automatisch für `AWS_CONTAINER_AUTHORIZATION_TOKEN` die Authentifizierung verwendet wird.
SDK for Java 1.x	Ja	Wenn Lambda aktiviert SnapStart ist `AWS_CONTAINER_CREDENTIALS_FULL_URI` und automatisch für `AWS_CONTAINER_AUTHORIZATION_TOKEN` die Authentifizierung verwendet wird.
SDK für 3.x JavaScript	Ja	
SDK für 2.x JavaScript	Ja	
SDK für Kotlin	Ja	
SDK for .NET 4.x	Ja	Wenn Lambda aktiviert SnapStart ist `AWS_CONTAINER_CREDENTIALS_FULL_URI` und automatisch für `AWS_CONTAINER_AUTHORIZATION_TOKEN` die Authentifizierung verwendet wird.
SDK for .NET 3.x	Ja	Wenn Lambda aktiviert SnapStart ist `AWS_CONTAINER_CREDENTIALS_FULL_URI` und automatisch für `AWS_CONTAINER_AUTHORIZATION_TOKEN` die Authentifizierung verwendet wird.
SDK for PHP 3.x	Ja	
SDK for Python (Boto3)	Ja	Wenn Lambda aktiviert SnapStart ist `AWS_CONTAINER_CREDENTIALS_FULL_URI` und automatisch für `AWS_CONTAINER_AUTHORIZATION_TOKEN` die Authentifizierung verwendet wird.
SDK for Ruby 3.x	Ja	
SDK für Rust	Ja	

SDK	Unterstützt	Hinweise oder weitere Informationen
SDK für Swift	Ja	
Tools für PowerShell V5	Ja	
Tools für V4 PowerShell	Ja	

IAM Identity Center-Anmeldeinformationsanbieter

> **Note**
>
> Hilfe zum Verständnis des Layouts von Einstellungsseiten oder zur Interpretation der nachfolgenden Tabelle Support by AWS SDKs and Tools finden Sie unterDie Einstellungsseiten dieses Handbuchs verstehen.

Dieser Authentifizierungsmechanismus wird verwendet AWS IAM Identity Center , um Single Sign-On (SSO) -Zugriff auf Ihren Code AWS-Services zu erhalten.

> **Note**
>
> In der AWS SDK-API-Dokumentation wird der IAM Identity Center-Anmeldeinformationsanbieter als SSO-Anmeldeinformationsanbieter bezeichnet.

Nachdem Sie IAM Identity Center aktiviert haben, definieren Sie ein Profil für die zugehörigen Einstellungen in Ihrer geteilten Datei. AWS `config` Dieses Profil wird verwendet, um eine Verbindung zum IAM Identity Center-Zugriffsportal herzustellen. Wenn sich ein Benutzer erfolgreich bei IAM Identity Center authentifiziert hat, gibt das Portal kurzfristige Anmeldeinformationen für die diesem Benutzer zugeordnete IAM-Rolle zurück. Informationen darüber, wie das SDK temporäre Anmeldeinformationen aus der Konfiguration erhält und sie für AWS-Service Anfragen verwendet, finden Sie unter. Wie die IAM Identity Center-Authentifizierung gelöst wird AWS SDKs und welche Tools

Es gibt zwei Möglichkeiten, IAM Identity Center über die `config` Datei zu konfigurieren:

- (Empfohlen) Konfiguration des SSO-Token-Anbieters — Verlängerte Sitzungsdauer. Beinhaltet Unterstützung für benutzerdefinierte Sitzungsdauern.
- Legacy-Konfiguration, die nicht aktualisiert werden kann — Verwendet eine feste, achtstündige Sitzung.

In beiden Konfigurationen müssen Sie sich erneut anmelden, wenn Ihre Sitzung abläuft.

Die folgenden beiden Leitfäden enthalten zusätzliche Informationen zu IAM Identity Center:

- AWS IAM Identity Center Benutzerhandbuch
- AWS IAM Identity Center Referenz zur Portal-API

Ausführliche Informationen darüber, wie die SDKs und Tools Anmeldeinformationen mithilfe dieser Konfiguration verwenden und aktualisieren, finden Sie unterWie die IAM Identity Center-Authentifizierung gelöst wird AWS SDKs und welche Tools.

Voraussetzungen

Sie müssen zuerst IAM Identity Center aktivieren. Einzelheiten zur Aktivierung der IAM Identity Center-Authentifizierung finden Sie unter Aktivieren AWS IAM Identity Center im AWS IAM Identity Center Benutzerhandbuch.

> **Note**
>
> Alternativ finden Sie die vollständigen Voraussetzungen und die erforderliche Konfiguration für gemeinsam genutzte `config` Dateien, die auf dieser Seite detailliert beschrieben werden, in der Anleitung zur EinrichtungVerwenden von IAM Identity Center zur Authentifizierung von AWS SDK und Tools.

Konfiguration des SSO-Token-Anbieters

Wenn Sie die Konfiguration des SSO-Token-Anbieters verwenden, aktualisiert Ihr AWS SDK oder Tool Ihre Sitzung automatisch bis zu Ihrem verlängerten Sitzungszeitraum. Weitere Informationen zur Sitzungsdauer und Höchstdauer finden Sie im Benutzerhandbuch unter Konfiguration der Sitzungsdauer des AWS Zugriffsportals und der integrierten IAM Identity AWS IAM Identity Center Center-Anwendungen.

Der `sso-session` Abschnitt der `config` Datei wird verwendet, um Konfigurationsvariablen für den Erwerb von SSO-Zugriffstoken zu gruppieren, die dann zum Abrufen von AWS Anmeldeinformationen verwendet werden können. Weitere Informationen zu diesem Abschnitt innerhalb einer `config` Datei finden Sie unterFormat der Konfigurationsdatei.

Im folgenden Beispiel für eine gemeinsam genutzte `config` Datei wird das SDK oder das Tool mithilfe eines `dev` Profils konfiguriert, um IAM Identity Center-Anmeldeinformationen anzufordern.

```
[profile dev]
sso_session = my-sso
sso_account_id = 111122223333
sso_role_name = SampleRole

[sso-session my-sso]
sso_region = us-east-1
sso_start_url = https://my-sso-portal.awsapps.com/start
sso_registration_scopes = sso:account:access
```

> **Note**
>
> Verwenden Sie für die AWS Regionsso_region, in der Ihre IAM Identity Center-Instanz konfiguriert ist. Dies kann von Ihrer AWS Standardregion abweichen. Sie finden diesen Wert im Dashboard der IAM Identity Center-Konsole.

Die vorherigen Beispiele zeigen, dass Sie einen `sso-session` Abschnitt definieren und ihn einem Profil zuordnen. In `sso_account_id` der Regel `sso_role_name` muss dies in dem `profile` Abschnitt festgelegt werden, damit das SDK AWS Anmeldeinformationen anfordern kann. `sso_regionsso_start_url`, und `sso_registration_scopes` muss innerhalb des `sso-session` Abschnitts festgelegt werden.

`sso_account_id`und `sso_role_name` sind nicht für alle Szenarien der SSO-Token-Konfiguration erforderlich. Wenn Ihre Anwendung nur AWS-Services diese Unterstützung für die Trägerauthentifizierung verwendet, sind herkömmliche AWS Anmeldeinformationen nicht erforderlich. Bei der Bearer-Authentifizierung handelt es sich um ein HTTP-Authentifizierungsschema, das Sicherheitstoken, sogenannte Bearer-Token, verwendet. In diesem Szenario sind `sso_account_id` und `sso_role_name` nicht erforderlich. In der jeweiligen AWS-Service Anleitung erfahren Sie, ob der Dienst die Bearer-Token-Autorisierung unterstützt.

Registrierungsbereiche werden als Teil eines konfiguriert. `sso-session` Der Geltungsbereich ist ein MechanismusOAuth 2.0, mit dem der Zugriff einer Anwendung auf das Konto eines Benutzers beschränkt wird. Im vorherigen Beispiel wird der erforderliche `sso_registration_scopes` Zugriff für die Auflistung von Konten und Rollen bereitgestellt.

Das folgende Beispiel zeigt, wie Sie dieselbe `sso-session` Konfiguration für mehrere Profile wiederverwenden können.

```
[profile dev]
sso_session = my-sso
sso_account_id = 111122223333
sso_role_name = SampleRole

[profile prod]
sso_session = my-sso
sso_account_id = 111122223333
sso_role_name = SampleRole2

[sso-session my-sso]
sso_region = us-east-1
sso_start_url = https://my-sso-portal.awsapps.com/start
sso_registration_scopes = sso:account:access
```

Das Authentifizierungstoken wird auf der Festplatte unter dem `~/.aws/sso/cache` Verzeichnis zwischengespeichert, wobei der Dateiname auf dem Sitzungsnamen basiert.

Nicht aktualisierbare Legacy-Konfiguration

Die automatisierte Token-Aktualisierung wird bei Verwendung der nicht aktualisierbaren Legacy-Konfiguration nicht unterstützt. Wir empfehlen, Konfiguration des SSO-Token-Anbieters stattdessen das zu verwenden.

> **Important**
>
> Bei der Legacy-Konfiguration ist Ihre Sitzung auf acht Stunden festgelegt und kann nicht automatisch aktualisiert werden. Wenn die Sitzung abläuft, schlägt jeder Code, der neue Anmeldeinformationen auflöst, bei der Authentifizierung fehl, bis Sie ihn erneut ausführen`aws sso login`. Wenn Sie über lange laufende Prozesse oder Automatisierungen verfügen, verwenden Sie die Konfiguration des SSO-Token-Anbieters, die die automatische Tokenaktualisierung unterstützt.

Um die alte, nicht aktualisierbare Konfiguration zu verwenden, müssen Sie die folgenden Einstellungen in Ihrem Profil angeben:

- `sso_start_url`
- `sso_region`
- `sso_account_id`
- `sso_role_name`

Sie geben das Benutzerportal für ein Profil mit den Einstellungen `sso_start_url` und `sso_region` an. Sie geben Berechtigungen mit den `sso_role_name` Einstellungen `sso_account_id` und an.

Im folgenden Beispiel werden die vier erforderlichen Werte in der `config` Datei festgelegt.

```
[profile my-sso-profile]
sso_start_url = https://my-sso-portal.awsapps.com/start
sso_region = us-west-2
sso_account_id = 111122223333
sso_role_name = SSOReadOnlyRole
```

Das Authentifizierungstoken wird auf der Festplatte unter dem `~/.aws/sso/cache` Verzeichnis zwischengespeichert, dessen Dateiname auf dem `sso_start_url` basiert.

Einstellungen des IAM Identity Center-Anmeldeinformationsanbieters

Konfigurieren Sie diese Funktionalität wie folgt:

`sso_start_url`- Einstellung für gemeinsam genutzte AWS **`config`** Dateien

Die URL, die auf die IAM Identity Center-Aussteller-URL oder die URL des Zugriffsportals Ihrer Organisation verweist. Weitere Informationen finden Sie im AWS IAM Identity Center Benutzerhandbuch unter Verwenden des AWS Zugriffsportals.

Um diesen Wert zu finden, öffnen Sie die IAM Identity Center-Konsole, sehen Sie sich das Dashboard an und suchen Sie nach der URL des AWS Zugriffsportals.

- Alternativ können Sie ab Version 2.22.0 von stattdessen den AWS CLI Wert für AWS die Aussteller-URL verwenden.

`sso_region`- Einstellung für gemeinsam genutzte Dateien AWS **`config`**

Die AWS-Region , die Ihren IAM Identity Center-Portalhost enthält, d. h. die Region, die Sie vor der Aktivierung von IAM Identity Center ausgewählt haben. Dies ist unabhängig von Ihrer AWS Standardregion und kann unterschiedlich sein.

Eine vollständige Liste der AWS-Regionen und ihrer Codes finden Sie unter Regionale Endpunkte in der Allgemeine Amazon Web Services-Referenz. Um diesen Wert zu finden, öffnen Sie die IAM Identity Center-Konsole, rufen Sie das Dashboard auf und suchen Sie nach Region.

`sso_account_id`- Einstellung für gemeinsam genutzte AWS **`config`** Dateien

Die numerische ID AWS-Konto , die über den AWS Organizations Dienst hinzugefügt wurde, um sie für die Authentifizierung zu verwenden.

Um die Liste der verfügbaren Konten zu sehen, gehen Sie zur IAM Identity Center-Konsole und öffnen Sie die AWS-KontenSeite. Die Liste der verfügbaren Konten, die die ListAccountsAPI-Methode verwenden, finden Sie auch in der AWS IAM Identity Center Portal-API-Referenz. Sie können beispielsweise die AWS CLI Methode list-accounts aufrufen.

`sso_role_name`- Einstellung für gemeinsam genutzte Dateien AWS **`config`**

Der Name eines als IAM-Rolle bereitgestellten Berechtigungssatzes, der die daraus resultierenden Berechtigungen des Benutzers definiert. Die Rolle muss in dem von AWS-Konto angegebenen Namen existieren. `sso_account_id` Verwenden Sie den Rollennamen, nicht den Rollennamen Amazon Resource Name (ARN).

Mit Berechtigungssätzen sind IAM-Richtlinien und benutzerdefinierte Berechtigungsrichtlinien verknüpft. Sie definieren die Zugriffsebene, die Benutzer auf die ihnen zugewiesenen AWS-Konten Rechte haben.

Um die Liste der verfügbaren Berechtigungssätze pro zu sehen AWS-Konto, gehen Sie zur IAM Identity Center-Konsole und öffnen Sie die AWS-KontenSeite. Wählen Sie den richtigen Namen für den Berechtigungssatz aus, der in der AWS-Konten Tabelle aufgeführt ist. Die Liste der verfügbaren Berechtigungssätze, die die ListAccountRolesAPI-Methode verwenden, finden Sie auch in der AWS IAM Identity Center Portal-API-Referenz. Sie können beispielsweise die AWS CLI Methode list-account-roles aufrufen.

`sso_registration_scopes`- Einstellung für gemeinsam genutzte Dateien AWS **`config`**

Eine durch Kommas getrennte Liste gültiger Bereichszeichenfolgen, für die autorisiert werden sollen. `sso-session` Eine Anwendung kann einen oder mehrere Bereiche anfordern und das an die Anwendung ausgegebene Zugriffstoken ist auf die gewährten Bereiche beschränkt. Ein

Mindestbereich von `sso:account:access` muss gewährt werden, um ein Aktualisierungstoken vom IAM Identity Center-Dienst zurückzuerhalten. Eine Liste der verfügbaren Optionen für den Zugriffsbereich finden Sie im AWS IAM Identity Center Benutzerhandbuch unter Zugriffsbereiche.

Diese Bereiche definieren die Berechtigungen, die für die Autorisierung für den registrierten OIDC-Client angefordert werden, und die vom Client abgerufenen Zugriffstoken. Bereiche autorisieren den Zugriff auf über IAM-Identity-Center-Bearer-Token autorisierte Endpunkte.

Diese Einstellung gilt nicht für die ältere, nicht aktualisierbare Konfiguration. Token, die mit der Legacy-Konfiguration ausgegeben wurden, sind implizit auf den Gültigkeitsbereich `sso:account:access` beschränkt.

Support durch AWS SDKs und Tools

Die folgenden SDKs unterstützen die in diesem Thema beschriebenen Funktionen und Einstellungen. Alle teilweisen Ausnahmen werden vermerkt. Alle Einstellungen für JVM-Systemeigenschaften werden AWS SDK für Kotlin nur von AWS SDK für Java und vom unterstützt.

SDK	Unterstützt	Hinweise oder weitere Informationen
AWS CLI v2	Ja	
SDK for C++	Ja	
SDK for Go V2 (1.x)	Ja	
SDK for Go 1.x (V1)	Ja	Um die Einstellungen für gemeinsam genutzte `config` Dateien zu verwenden, müssen Sie das Laden aus der Konfigurationsdatei aktivieren. Weitere Informationen finden Sie unter Sessions.
SDK for Java 2.x	Ja	Konfigurationswerte werden auch in der `credentials` Datei unterstützt.
SDK for Java 1.x	Nein	
SDK für 3.x JavaScript	Ja	

SDK	Unterstützt	Hinweise oder weitere Informationen
SDK für 2.x JavaScript	Ja	
SDK für Kotlin	Ja	
SDK for .NET 4.x	Ja	
SDK for .NET 3.x	Ja	
SDK for PHP 3.x	Ja	
SDK for Python (Boto3)	Ja	
SDK for Ruby 3.x	Ja	
SDK für Rust	Teilweise	Nur ältere, nicht aktualisierbare Konfiguration.
SDK für Swift	Ja	
Tools für PowerShell V5	Ja	
Tools für V4 PowerShell	Ja	

Note

Hier sind die relevanten Dokumente für AWS Toolkits und die IAM Identity Center-Authentifizierung:

- VS Code — Verbindung herstellen mit AWS— Deckt die IAM Identity Center-Anmeldung über die Start-URL und die browserbasierte Authentifizierung ab.
- Visual Studio — AWS IAM Identity Center-Anmeldeinformationen im AWS Toolkit for Visual Studio — Behandelt die Konfiguration von SSO-Profilen.
- JetBrains— Das AWS Toolkit für mit Ihrem AWS Konto JetBrains verbinden — Erläutert, wie Sie das AWS Toolkit für JetBrains mit Ihrem Konto verbinden. AWS

IMDS-Anmeldeinformationsanbieter

> **Note**
>
> Hilfe zum Verständnis des Layouts von Einstellungsseiten oder zur Interpretation der nachfolgenden Tabelle Support by AWS SDKs and Tools finden Sie unterDie Einstellungsseiten dieses Handbuchs verstehen.

Der Instanz-Metadatendienst (IMDS) stellt Daten über Ihre Instance bereit, die Sie zur Konfiguration oder Verwaltung der laufenden Instance verwenden können. Weitere Informationen zu den verfügbaren Daten finden Sie unter Arbeiten mit Instance-Metadaten im Amazon EC2 EC2-Benutzerhandbuch. Amazon EC2 bietet einen lokalen Endpunkt, der Instances zur Verfügung steht und der Instance verschiedene Informationen zur Verfügung stellen kann. Wenn der Instance eine Rolle zugewiesen ist, kann sie eine Reihe von Anmeldeinformationen bereitstellen, die für diese Rolle gültig sind. Die SDKs können diesen Endpunkt verwenden, um Anmeldeinformationen als Teil ihrer standardmäßigen Anbieterkette für Anmeldeinformationen aufzulösen. Standardmäßig wird Instance Metadata Service Version 2 (IMDSv2) verwendet, eine sicherere Version von IMDS, die ein Sitzungstoken verwendet. Wenn dies aufgrund eines Zustands fehlschlägt, der nicht erneut versucht werden kann (HTTP-Fehlercodes 403, 404, 405), wird IMDSv1 als Fallback verwendet.

Konfigurieren Sie diese Funktionalität wie folgt:

`AWS_EC2_METADATA_DISABLED`- Umgebungsvariable

Ob versucht werden soll, den Amazon EC2 Instance Metadata Service (IMDS) zum Abrufen von Anmeldeinformationen zu verwenden.

Standardwert: `false`.

Zulässige Werte:

- **`true`**— Verwenden Sie IMDS nicht, um Anmeldeinformationen zu erhalten.
- **`false`**— Verwenden Sie IMDS, um Anmeldeinformationen zu erhalten.

`ec2_metadata_v1_disabled`- Einstellung für gemeinsam genutzte AWS **`config`** Dateien, **`AWS_EC2_METADATA_V1_DISABLED`**- Umgebungsvariable, **`aws.disableEc2MetadataV1`**- JVM-Systemeigenschaft: nur Java/Kotlin

Gibt an, ob Instance Metadata Service Version 1 (IMDSv1) als Fallback verwendet werden soll, falls IMDSv2 fehlschlägt.

> **Note**
>
> Neue SDKs unterstützen IMDSv1 nicht und unterstützen daher diese Einstellung nicht. Einzelheiten finden Sie in der Tabelle. Support durch AWS SDKs und Tools

Standardwert: `false`.

Zulässige Werte:

- **`true`**— Verwenden Sie IMDSv1 nicht als Fallback.
- **`false`**— Verwenden Sie IMDSv1 als Fallback.

`ec2_metadata_service_endpoint`- Einstellung für gemeinsam genutzte Dateien AWS **`config`**, **`AWS_EC2_METADATA_SERVICE_ENDPOINT`**- Umgebungsvariable, **`aws.ec2MetadataServiceEndpoint`**- JVM-Systemeigenschaft: nur Java/Kotlin

Der Endpunkt von IMDS. Dieser Wert überschreibt den Standardspeicherort, an dem AWS SDKs und Tools nach Amazon EC2 EC2-Instance-Metadaten suchen.

Standardwert: Wenn `ec2_metadata_service_endpoint_mode` gleich, dann ist der `IPv4` Standardendpunkt. `http://169.254.169.254` Wenn `ec2_metadata_service_endpoint_mode` gleich`IPv6`, dann ist der Standardendpunkt. `http://[fd00:ec2::254]`

Gültige Werte: Gültiger URI.

`ec2_metadata_service_endpoint_mode`- Einstellung für gemeinsam genutzte AWS **`config`** Dateien, **`AWS_EC2_METADATA_SERVICE_ENDPOINT_MODE`**- Umgebungsvariable, **`aws.ec2MetadataServiceEndpointMode`**- JVM-Systemeigenschaft: nur Java/Kotlin

Der Endpunktmodus von IMDS.

Standardwert:`IPv4`.

Gültige Werte:`IPv4`,`IPv6`.

> **Note**
>
> Der IMDS-Anmeldeinformationsanbieter ist Teil von. Machen Sie sich mit der Kette der Anbieter von Anmeldeinformationen vertraut Der IMDS-Anmeldeinformationsanbieter wird jedoch erst nach mehreren anderen Anbietern dieser Serie überprüft. Wenn Sie also

möchten, dass Ihr Programm die Anmeldeinformationen dieses Anbieters verwendet, müssen Sie andere gültige Anmeldeinformationsanbieter aus Ihrer Konfiguration entfernen oder ein anderes Profil verwenden. Anstatt sich auf die Kette der Anmeldeinformationsanbieter zu verlassen, um automatisch zu ermitteln, welcher Anbieter gültige Anmeldeinformationen zurückgibt, können Sie alternativ die Verwendung des IMDS-Anmeldeinformationsanbieters im Code angeben. Sie können die Quellen für Anmeldeinformationen direkt angeben, wenn Sie Service-Clients erstellen.

Sicherheit für IMDS-Anmeldeinformationen

Wenn das AWS SDK nicht mit gültigen Anmeldeinformationen konfiguriert ist, versucht das SDK standardmäßig, den Amazon EC2 Instance Metadata Service (IMDS) zu verwenden, um Anmeldeinformationen für eine AWS Rolle abzurufen. Dieses Verhalten kann deaktiviert werden, indem die `AWS_EC2_METADATA_DISABLED` Umgebungsvariable auf gesetzt wird. `true` Dies verhindert unnötige Netzwerkaktivitäten und erhöht die Sicherheit in nicht vertrauenswürdigen Netzwerken, in denen der Amazon EC2 Instance Metadata Service möglicherweise imitiert wird.

> **Note**
>
> AWS SDK-Clients, die mit gültigen Anmeldeinformationen konfiguriert sind, verwenden IMDS niemals, um Anmeldeinformationen abzurufen, unabhängig von diesen Einstellungen.

Verwendung von Amazon EC2 IMDS-Anmeldeinformationen deaktivieren

Wie Sie diese Umgebungsvariable festlegen, hängt davon ab, welches Betriebssystem verwendet wird und ob die Änderung dauerhaft sein soll oder nicht.

Unter Linux und macOS

Kunden, die Linux oder macOS verwenden, können diese Umgebungsvariable mit dem folgenden Befehl festlegen:

```
$ export AWS_EC2_METADATA_DISABLED=true
```

Wenn Sie möchten, dass diese Einstellung über mehrere Shell-Sitzungen und Systemneustarts hinweg beibehalten wird, können Sie den obigen Befehl zu Ihrer Shell-Profildatei hinzufügen, z. B. `.bash_profile` `.zsh_profile`, oder `.profile`.

Windows

Kunden, die Windows verwenden, können diese Umgebungsvariable mit dem folgenden Befehl festlegen:

```
$ set AWS_EC2_METADATA_DISABLED=true
```

Wenn Sie möchten, dass diese Einstellung über mehrere Shell-Sitzungen und Systemneustarts hinweg erhalten bleibt, können Sie stattdessen den folgenden Befehl verwenden:

```
$ setx AWS_EC2_METADATA_DISABLED=true
```

> **Note**
>
> Der setx Befehl wendet den Wert nicht auf die aktuelle Shell-Sitzung an, sodass Sie die Shell neu laden oder erneut öffnen müssen, damit die Änderung wirksam wird.

Support durch AWS SDKs und Tools

Die folgenden SDKs unterstützen die in diesem Thema beschriebenen Funktionen und Einstellungen. Alle teilweisen Ausnahmen werden vermerkt. Alle Einstellungen für JVM-Systemeigenschaften werden AWS SDK für Kotlin nur von AWS SDK für Java und vom unterstützt.

SDK	Unterstützt	Hinweise oder weitere Informationen
AWS CLI v2	Ja	
SDK for C++	Ja	
SDK for Go V2 (1.x)	Ja	
SDK for Go 1.x (V1)	Ja	Um die Einstellungen für gemeinsam genutzte `config` Dateien zu verwenden, müssen Sie das Laden aus der Konfigurationsdatei aktivieren. Weitere Informationen finden Sie unter Sessions.
SDK for Java 2.x	Ja	

SDK	Unterstützt	Hinweise oder weitere Informationen
SDK for Java 1.x	Teilweise	JVM-Systemeigenschaften: Wird `com.amazonaws.sdk.disableEc2MetadataV1` anstelle von `aws.disableEc2MetadataV1`; verwendet `aws.ec2MetadataServiceEndpoint` und wird `aws.ec2MetadataServiceEndpointMode` nicht unterstützt.
SDK für 3.x JavaScript	Ja	
SDK für 2.x JavaScript	Ja	
SDK für Kotlin	Ja	Verwendet kein IMDSv1-Fallback.
SDK for .NET 4.x	Ja	
SDK for .NET 3.x	Ja	
SDK for PHP 3.x	Ja	
SDK for Python (Boto3)	Ja	
SDK for Ruby 3.x	Ja	
SDK für Rust	Ja	Verwendet kein IMDSv1-Fallback.
SDK für Swift	Ja	
Tools für PowerShell V5	Ja	Sie können den IMDSv1-Fallback explizit im Code deaktivieren, indem Sie. `[Amazon.Util.EC2InstanceMetadata]::EC2MetadataV1Disabled = $true`
Tools für V4 PowerShell	Ja	Sie können den IMDSv1-Fallback explizit im Code mithilfe von deaktivieren. `[Amazon.Util.EC2InstanceMetadata]::EC2MetadataV1Disabled = $true`

Anbieter von Prozessanmeldeinformationen

> **Note**
>
> Hilfe zum Verständnis des Layouts von Einstellungsseiten oder zur Interpretation der nachfolgenden Tabelle Support by AWS SDKs and Tools finden Sie unterDie Einstellungsseiten dieses Handbuchs verstehen.

SDKs bieten eine Möglichkeit, die Kette der Anbieter von Anmeldeinformationen für benutzerdefinierte Anwendungsfälle zu erweitern. Dieser Anbieter kann verwendet werden, um benutzerdefinierte Implementierungen bereitzustellen, z. B. das Abrufen von Anmeldeinformationen aus einem lokalen Anmeldeinformationsspeicher oder die Integration mit Ihrem lokalen Identitätsanbieter.

Zum Beispiel verwendet IAM Roles Anywhere, `credential_process` um temporäre Anmeldeinformationen für Ihre Anwendung abzurufen. Informationen zur Konfiguration `credential_process` für diese Verwendung finden Sie unterAuthentifizierung AWS SDKs und Tools mit IAM Roles Anywhere.

> **Important**
>
> Die `credential_process` Einstellung führt ein externes Programm auf Ihrem System aus, um Anmeldeinformationen abzurufen. Stellen Sie sicher, dass Sie dem in dieser Einstellung angegebenen Programm vertrauen, da jeder, der Schreibzugriff auf Ihre gemeinsam genutzte `config` Datei hat, den ausgeführten Befehl ändern kann. Schützen Sie Ihre `config` Datei mit Dateisystemberechtigungen, die für Ihr Betriebssystem geeignet sind.

> **Note**
>
> Im Folgenden wird eine Methode zum Abrufen von Anmeldeinformationen aus einem externen Prozess beschrieben. Diese Methode kann verwendet werden, wenn Sie Software außerhalb von AWS ausführen. Wenn Sie auf einer AWS Rechenressource aufbauen, verwenden Sie andere Anbieter von Anmeldeinformationen. Wenn Sie diese Option verwenden, sollten Sie sicherstellen, dass die Konfigurationsdatei so gesperrt wie möglich ist. Verwenden Sie dabei bewährte Sicherheitsmethoden für Ihr Betriebssystem. Stellen Sie sicher, dass Ihr benutzerdefiniertes Anmeldeinformationstool keine geheimen Informationen

speichert`StdErr`, da die SDKs solche Informationen erfassen und protokollieren AWS CLI können, wodurch sie möglicherweise unbefugten Benutzern zugänglich gemacht werden.

Konfigurieren Sie diese Funktionalität wie folgt:

`credential_process`- Einstellung für gemeinsam genutzte AWS **`config`** Dateien

Gibt einen externen Befehl an, den das SDK oder Tool in Ihrem Namen ausführt, um zu verwendende Authentifizierungsdaten zu generieren oder abzurufen. Die Einstellung gibt den Namen einer an program/command , die das SDK aufrufen wird. Wenn das SDK den Prozess aufruft, wartet es darauf, dass der Prozess JSON-Daten schreibt. `stdout` Der benutzerdefinierte Anbieter muss Informationen in einem bestimmten Format zurückgeben. Diese Informationen enthalten die Anmeldeinformationen, mit denen das SDK oder das Tool Sie authentifizieren kann.

> **Note**
>
> Der Anbieter von Prozessanmeldedaten ist Teil von. Machen Sie sich mit der Kette der Anbieter von Anmeldeinformationen vertraut Der Anbieter für Prozessanmeldedaten wird jedoch erst nach mehreren anderen Anbietern dieser Serie überprüft. Wenn Sie also möchten, dass Ihr Programm die Anmeldeinformationen dieses Anbieters verwendet, müssen Sie andere gültige Anmeldeinformationsanbieter aus Ihrer Konfiguration entfernen oder ein anderes Profil verwenden. Anstatt sich auf die Kette der Anmeldeinformationsanbieter zu verlassen, um automatisch zu ermitteln, welcher Anbieter gültige Anmeldeinformationen zurückgibt, können Sie alternativ die Verwendung des Anbieters für Prozessanmeldedaten im Code angeben. Sie können die Quellen für Anmeldeinformationen direkt angeben, wenn Sie Dienstclients erstellen.

Den Pfad zum Programm für Anmeldeinformationen angeben

Der Wert der Einstellung ist eine Zeichenfolge, die einen Pfad zu einem Programm enthält, das das SDK oder das Entwicklungstool in Ihrem Namen ausführt:

- Der Pfad und der Dateiname dürfen nur aus den folgenden Zeichen bestehen: A-Z, a-z, 0-9, Bindestrich (-), Unterstrich (_), Punkt (.), Schrägstrich (/), umgekehrter Schrägstrich (\) und Leerzeichen.

- Wenn der Pfad oder Dateiname ein Leerzeichen enthält, umgeben Sie den vollständigen Pfad und Dateinamen mit doppelten Anführungszeichen („ “).
- Wenn ein Parametername oder ein Parameterwert ein Leerzeichen enthält, umgeben Sie dieses Element mit doppelten Anführungszeichen („ “). Umgeben Sie dabei nur den Namen oder den Wert, nicht beides.
- Nehmen Sie keine Umgebungsvariablen in die Zeichenketten auf. Fügen Sie beispielsweise `$HOME` oder nicht ein`%USERPROFILE%`.
- Geben Sie den Basisordner nicht als an~. * Sie müssen entweder den vollständigen Pfad oder einen Basisdateinamen angeben. Wenn es einen Basisdateinamen gibt, versucht das System, das Programm in den durch die `PATH` Umgebungsvariable angegebenen Ordnern zu finden. Der Pfad variiert je nach Betriebssystem:

  Das folgende Beispiel zeigt, dass credentials al_process in der gemeinsam genutzten `config` Datei aktiviert ist. Linux/macOS

  ```
  credential_process = "/path/to/credentials.sh" parameterWithoutSpaces "parameter with
   spaces"
  ```

  Das folgende Beispiel zeigt die Einstellung von credentials al_process in der gemeinsam genutzten Datei unter Windows. `config`

  ```
  credential_process = "C:\Path\To\credentials.cmd" parameterWithoutSpaces "parameter
   with spaces"
  ```

- Kann in einem speziellen Profil angegeben werden:

  ```
  [profile cred_process]
  credential_process = /Users/username/process.sh
  region = us-east-1
  ```

Gültige Ausgabe des Anmeldeinformationsprogramms

Das SDK führt den Befehl wie im Profil angegeben aus und liest dann Daten aus dem Standardausgabestream. Der von Ihnen angegebene Befehl, unabhängig davon, ob es sich um ein Skript oder ein Binärprogramm handelt, muss eine JSON-Ausgabe generieren`STDOUT`, die der folgenden Syntax entspricht.

```
{
    "Version": 1,
    "AccessKeyId": "an AWS access key",
    "SecretAccessKey": "your AWS secret access key",
    "SessionToken": "the AWS session token for temporary credentials",
    "Expiration": "RFC3339 timestamp for when the credentials expire",
    "AccountId": "the AWS account ID associated with these credentials"
}
```

> **Note**
>
> Derzeit muss der `Version`-Schlüssel auf 1 gesetzt sein. Im Laufe der Zeit kann ein höherer Wert erforderlich sein, wenn sich die Struktur weiterentwickelt.

Der `Expiration` Schlüssel ist ein Zeitstempel im RFC3339-Format. Wenn der `Expiration` Schlüssel nicht in der Ausgabe des Tools enthalten ist, geht das SDK davon aus, dass es sich bei den Anmeldeinformationen um langfristige Anmeldeinformationen handelt, die nicht aktualisiert werden. Andernfalls werden die Anmeldeinformationen als temporäre Anmeldeinformationen betrachtet und sie werden automatisch aktualisiert, indem der `credential_process` Befehl erneut ausgeführt wird, bevor die Anmeldeinformationen ablaufen.

> **Note**
>
> Das SDK speichert die Anmeldeinformationen für externe Prozesse nicht im Cache, so wie es bei der Übernahme von Rollenanmeldedaten der Fall ist. Wenn Caching erforderlich ist, müssen Sie dies im externen Prozess implementieren.

> **Note**
>
> `AccountId`dist optional, sodass Anmeldeinformationen auch ohne sie funktionieren, aber ihre Bereitstellung ermöglicht eine optimierte Endpunktauflösung für Dienste, die sie unterstützen. Wenn das SDK weiß, zu welchem Konto die Anmeldeinformationen gehören, können bestimmte AWS Dienste kontospezifische Endpunkte verwenden. Beispielsweise kann Amazon S3 Anfragen an einen kontobezogenen Endpunkt und nicht an den globalen Endpunkt weiterleiten. `AccountId`Andernfalls muss sich das SDK auf andere Mechanismen

verlassen, um das Konto zu ermitteln, oder es kann kein kontobasiertes Endpunkt-Routing verwenden.

Der externe Prozess kann einen Rückgabecode ungleich Null zurückgeben, um anzuzeigen, dass beim Abrufen der Anmeldeinformationen ein Fehler aufgetreten ist.

Support durch AWS SDKs und Tools

Die folgenden SDKs unterstützen die in diesem Thema beschriebenen Funktionen und Einstellungen. Alle teilweisen Ausnahmen werden vermerkt. Alle Einstellungen für JVM-Systemeigenschaften werden AWS SDK für Kotlin nur von AWS SDK für Java und vom unterstützt.

SDK	Unterstützt	Hinweise oder weitere Informationen
AWS CLI v2	Ja	
SDK for C++	Ja	
SDK for Go V2 (1.x)	Ja	
SDK for Go 1.x (V1)	Ja	Um die Einstellungen für gemeinsam genutzte `config` Dateien zu verwenden, müssen Sie das Laden aus der Konfigurationsdatei aktivieren; siehe Sessions.
SDK for Java 2.x	Ja	
SDK for Java 1.x	Ja	
SDK für 3.x JavaScript	Ja	
SDK für 2.x JavaScript	Ja	
SDK für Kotlin	Ja	
SDK for .NET 4.x	Ja	
SDK for .NET 3.x	Ja	

SDK	Unterstützt	Hinweise oder weitere Informationen
SDK for PHP 3.x	Ja	
SDK for Python (Boto3)	Ja	
SDK for Ruby 3.x	Ja	
SDK für Rust	Ja	
SDK für Swift	Ja	
Tools für PowerShell V5	Ja	
Tools für V4 PowerShell	Ja	

AWS Standardisierte Funktionen von SDKs und Tools

Viele Funktionen wurden auf einheitliche Standardwerte standardisiert und funktionieren in vielen SDKs auf die gleiche Weise. Diese Konsistenz erhöht die Produktivität und Klarheit bei der Codierung mehrerer SDKs. Alle Einstellungen können im Code überschrieben werden. Einzelheiten finden Sie in Ihrer spezifischen SDK-API.

> **Important**
>
> Nicht alle SDKs unterstützen alle Funktionen oder sogar alle Aspekte innerhalb einer Funktion.

Themen

- Account-based Endpunkte
- Anwendungs-ID
- Amazon EC2-Instance-Metadaten
- Amazon S3 Access Points
- Amazon S3 Multi-Region S3-Zugriffspunkte
- S3 Express One Zone-Sitzungsauthentifizierung

- Authentifizierungsschema
- AWS-Region
- AWS STS Regionale -Endpunkte
- Schutz der Datenintegrität für Amazon S3
- Dual-stack und FIPS-Endpunkte
- Endpunkterkennung
- Allgemeine Konfigurationseinstellungen
- Host-Präfix-Injektion
- IMDS-Kunde
- Wiederholungsverhalten
- Komprimierung anfordern
- Service-specific Endpunkte
- Standardeinstellungen für intelligente Konfigurationen
- Paginierung

Account-based Endpunkte

> **Note**
>
> Hilfe zum Verständnis des Layouts von Einstellungsseiten oder zur Interpretation der nachfolgenden Tabelle Support by AWS SDKs and Tools finden Sie unterDie Einstellungsseiten dieses Handbuchs verstehen.

Account-based Endgeräte tragen dazu bei, eine hohe Leistung und Skalierbarkeit sicherzustellen, indem sie Ihre AWS-Konto ID verwenden, um Anfragen für Dienste weiterzuleiten, die diese Funktion unterstützen. Wenn Sie ein AWS SDK und einen Dienst verwenden, die kontobasierte Endpunkte unterstützen, erstellt und verwendet der SDK-Client einen kontobasierten Endpunkt anstelle eines regionalen Endpunkts. Wenn die Konto-ID für den SDK-Client nicht sichtbar ist, verwendet der Client den regionalen Endpunkt. Account-basedEndpunkte haben die Form von`https://<account-id>.ddb.<region>.amazonaws.com`, wo `<account-id>` und `<region>` sind Ihre AWS-Konto ID und AWS-Region.

Konfigurieren Sie diese Funktionalität wie folgt:

`aws_account_id`- Einstellung für gemeinsam genutzte AWS **`config`** Dateien, **`AWS_ACCOUNT_ID`**- Umgebungsvariable, **`aws.accountId`**- JVM-Systemeigenschaft: nur Java/Kotlin

Die AWS-Konto ID. Wird für kontobasiertes Endpunkt-Routing verwendet. Eine AWS-Konto ID hat ein Format wie 111122223333.

Account-based Das Endpunkt-Routing bietet für einige Dienste eine bessere Anforderungsleistung.

`account_id_endpoint_mode`- Einstellung für gemeinsam genutzte AWS **`config`** Dateien, **`AWS_ACCOUNT_ID_ENDPOINT_MODE`**- Umgebungsvariable, **`aws.accountIdEndpointMode`**- JVM-Systemeigenschaft: nur Java/Kotlin

Diese Einstellung wird verwendet, um das kontobasierte Endpunkt-Routing bei Bedarf zu deaktivieren und kontobasierte Regeln zu umgehen.

Standardwert: `preferred`

Zulässige Werte:

- **`preferred`**— Der Endpunkt sollte die Konto-ID enthalten, falls verfügbar.
- **`disabled`** – Ein aufgelöster Endpunkt enthält keine Konto-ID.
- **`required`** – Der Endpunkt muss die Konto-ID enthalten. Wenn die Konto-ID nicht verfügbar ist, gibt das SDK einen Fehler aus.

Support durch AWS SDKs und Tools

Die folgenden SDKs unterstützen die in diesem Thema beschriebenen Funktionen und Einstellungen. Alle teilweisen Ausnahmen werden vermerkt. Alle Einstellungen für JVM-Systemeigenschaften werden AWS SDK für Kotlin nur von AWS SDK für Java und vom unterstützt.

SDK	Unterzt	In der SDK-Version veröffentlicht	Hinweise oder weitere Informationen
AWS CLI v2	Ja	2.25.0	
AWS CLI v1	Ja	1.38,0	

SDK	Unterzt	In der SDK-Version veröffentlicht	Hinweise oder weitere Informationen
SDK for C++	Nein		
SDK for Go V2 (1.x)	Ja	v1.35.0	
SDK for Go 1.x (V1)	Nein		
SDK for Java 2.x	Ja	v2.28.4	
SDK for Java 1.x	Ja	v1.12.771	
SDK für 3.x JavaScript	Ja	v3.656.0	
SDK für 2.x JavaScript	Nein		
SDK für Kotlin	Ja	v1.3.37	
SDK for .NET 4.x	Ja	4.0.0	
SDK for .NET 3.x	Nein		
SDK for PHP 3.x	Ja	v3.318.0	
SDK for Python (Boto3)	Ja	1.37.0	
SDK for Ruby 3.x	Ja	v1.123.0	
SDK für Rust	Ja	Veröffentlichung-2025-04-24	
SDK für Swift	Ja	1.2.0	

SDK	Unterzt	In der SDK-Version veröffentlicht	Hinweise oder weitere Informationen
Tools für PowerShell V5	Nein		
Tools für V4 PowerShell	Nein		

Anwendungs-ID

> **Note**
>
> Hilfe zum Verständnis des Layouts von Einstellungsseiten oder zur Interpretation der nachfolgenden Tabelle Support by AWS SDKs and Tools finden Sie unterDie Einstellungsseiten dieses Handbuchs verstehen.

Eine einzelne AWS-Konto kann von mehreren Kundenanwendungen verwendet werden, um Anrufe zu AWS-Services tätigen. Mithilfe der Anwendungs-ID können Kunden ermitteln, welche Quellanwendung eine Reihe von Aufrufen mithilfe von getätigt hat AWS-Konto. AWS SDKs und Dienste verwenden oder interpretieren diesen Wert nur, um ihn in der Kundenkommunikation wieder auftauchen zu lassen. Dieser Wert kann beispielsweise in betrieblichen E-Mails oder in der enthalten sein, AWS Health Dashboard um eindeutig zu identifizieren, welche Ihrer Anwendungen mit der Benachrichtigung verknüpft ist.

Konfigurieren Sie diese Funktionalität wie folgt:

`sdk_ua_app_id`- Einstellung für gemeinsam genutzte AWS **`config`** Dateien, **`AWS_SDK_UA_APP_ID`**- Umgebungsvariable, **`sdk.ua.appId`**- JVM-Systemeigenschaft: nur Java/Kotlin

Diese Einstellung ist eine eindeutige Zeichenfolge, die Sie Ihrer Anwendung zuweisen, um zu identifizieren, welche Ihrer Anwendungen innerhalb einer bestimmten Anwendung Aufrufe AWS-Konto tätigt. AWS

Standardwert: None

Gültige Werte: Zeichenfolge mit einer maximalen Länge von 50. Buchstaben, Zahlen und die folgenden Sonderzeichen sind zulässig: !,#,,$,%,&,',*,+,-,.,^,_,`,|,~.

Beispiel für die Einstellung dieses Werts in der `config` Datei:

```
[default]
sdk_ua_app_id=ABCDEF
```

Linux/macOS Beispiel für das Setzen von Umgebungsvariablen über die Befehlszeile:

```
export AWS_SDK_UA_APP_ID=ABCDEF
export AWS_SDK_UA_APP_ID="ABC DEF"
```

Windows-Beispiel für das Setzen von Umgebungsvariablen über die Befehlszeile:

```
setx AWS_SDK_UA_APP_ID ABCDEF
setx AWS_SDK_UA_APP_ID="ABC DEF"
```

Wenn Sie Symbole verwenden, die für die verwendete Shell eine besondere Bedeutung haben, maskieren Sie den Wert entsprechend.

Support durch AWS SDKs und Tools

Die folgenden SDKs unterstützen die in diesem Thema beschriebenen Funktionen und Einstellungen. Alle teilweisen Ausnahmen werden vermerkt. Alle Einstellungen für JVM-Systemeigenschaften werden AWS SDK für Kotlin nur von AWS SDK für Java und vom unterstützt.

SDK	Unterstützt	Hinweise oder weitere Informationen
AWS CLI v2	Ja	
SDK for C++	Ja	Die gemeinsam genutzte `config` Datei wird nicht unterstützt.
SDK for Go V2 (1.x)	Ja	
SDK for Go 1.x (V1)	Nein	

SDK	Unterstützt	Hinweise oder weitere Informationen
SDK for Java 2.x	Ja	
SDK for Java 1.x	Nein	
SDK für 3.x JavaScript	Ja	
SDK für 2.x JavaScript	Nein	
SDK für Kotlin	Ja	Die JVM-Systemeigenschaft ist. `aws.userAgentAppId`
SDK for .NET 4.x	Ja	
SDK for .NET 3.x	Ja	
SDK for PHP 3.x	Ja	
SDK for Python (Boto3)	Ja	
SDK for Ruby 3.x	Ja	
SDK für Rust	Ja	
SDK für Swift	Ja	
Tools für PowerShell V5	Ja	
Tools für V4 PowerShell	Ja	

Amazon EC2-Instance-Metadaten

> **Note**
>
> Hilfe zum Verständnis des Layouts von Einstellungsseiten oder zur Interpretation der nachfolgenden Tabelle Support by AWS SDKs and Tools finden Sie unterDie Einstellungsseiten dieses Handbuchs verstehen.

Amazon EC2 bietet einen Service für Instances, den sogenannten Instance Metadata Service (IMDS). Weitere Informationen zu diesem Service finden Sie unter Arbeiten mit Instance-Metadaten im Amazon EC2 EC2-Benutzerhandbuch. Beim Versuch, Anmeldeinformationen auf einer Amazon EC2 EC2-Instance abzurufen, die mit einer IAM-Rolle konfiguriert wurde, ist die Verbindung zum Instance-Metadaten-Service anpassbar.

Konfigurieren Sie diese Funktionalität wie folgt:

`metadata_service_num_attempts`- Einstellung für gemeinsam genutzte AWS **`config`** Dateien, **`AWS_METADATA_SERVICE_NUM_ATTEMPTS`**- Umgebungsvariable

Diese Einstellung gibt die Gesamtzahl der Versuche an, die unternommen werden müssen, bevor der Versuch, Daten aus dem Instanz-Metadatendienst abzurufen, aufgegeben wird.

Standardwert: 1

Gültige Werte: Zahl größer oder gleich 1.

`metadata_service_timeout`- Einstellung für gemeinsam genutzte AWS **`config`** Dateien, **`AWS_METADATA_SERVICE_TIMEOUT`**- Umgebungsvariable

Gibt die Anzahl der Sekunden an, bevor beim Versuch, Daten vom Instanz-Metadatendienst abzurufen, ein Timeout eintritt.

Standardwert: 1

Gültige Werte: Zahl größer oder gleich 1.

Beispiel für das Einstellen dieser Werte in der `config` Datei:

```
[default]
metadata_service_num_attempts=10
metadata_service_timeout=10
```

Linux/macOS Beispiel für das Setzen von Umgebungsvariablen über die Befehlszeile:

```
export AWS_METADATA_SERVICE_NUM_ATTEMPTS=10
export AWS_METADATA_SERVICE_TIMEOUT=10
```

Windows-Beispiel für das Setzen von Umgebungsvariablen über die Befehlszeile:

```
setx AWS_METADATA_SERVICE_NUM_ATTEMPTS 10
```

```
setx AWS_METADATA_SERVICE_TIMEOUT 10
```

Support durch AWS SDKs und Tools

Die folgenden SDKs unterstützen die in diesem Thema beschriebenen Funktionen und Einstellungen. Alle teilweisen Ausnahmen werden vermerkt. Alle Einstellungen für JVM-Systemeigenschaften werden AWS SDK für Kotlin nur von AWS SDK für Java und vom unterstützt.

SDK	Unterstützt	Hinweise oder weitere Informationen
AWS CLI v2	Ja	
SDK for C++	Nein	
SDK for Go V2 (1.x)	Nein	
SDK for Go 1.x (V1)	Nein	
SDK for Java 2.x	Teilweise	Nur `AWS_METADATA_SERVICE_TIMEOUT` wird unterstützt.
SDK for Java 1.x	Teilweise	Nur `AWS_METADATA_SERVICE_TIMEOUT` wird unterstützt.
SDK für 3.x JavaScript	Nein	
SDK für 2.x JavaScript	Nein	
SDK für Kotlin	Nein	
SDK for .NET 4.x	Nein	
SDK for .NET 3.x	Nein	
SDK for PHP 3.x	Ja	
SDK for Python (Boto3)	Ja	
SDK for Ruby 3.x	Nein	

SDK	U zt	Hinweise oder weitere Informationen
SDK für Rust	Neir	
SDK für Swift	Neir	
Tools für PowerShell V5	Neir	
Tools für V4 PowerShell	Neir	

Amazon S3 Access Points

> **Note**
>
> Hilfe zum Verständnis des Layouts von Einstellungsseiten oder zur Interpretation der nachfolgenden Tabelle Support by AWS SDKs and Tools finden Sie unterDie Einstellungsseiten dieses Handbuchs verstehen.

Der Amazon S3 S3-Service bietet Access Points als alternative Möglichkeit zur Interaktion mit Amazon S3 S3-Buckets. Access Points verfügen über einzigartige Richtlinien und Konfigurationen, die auf sie angewendet werden können, anstatt direkt auf den Bucket. Mit AWS SDKs können Sie Access Point Amazon Resource Names (ARNs) im Bucket-Feld für API-Operationen verwenden, anstatt den Bucket-Namen explizit anzugeben. Sie werden für bestimmte Operationen verwendet, z. B. die Verwendung eines Access Point-ARN `GetObject`zum Abrufen eines Objekts aus einem Bucket oder die Verwendung eines Access Point-ARN mit, `PutObject`um einem Bucket ein Objekt hinzuzufügen.

Weitere Informationen zu Amazon S3 S3-Zugriffspunkten und ARNs finden Sie unter Verwenden von Zugriffspunkten im Amazon S3 S3-Benutzerhandbuch.

Konfigurieren Sie diese Funktionalität wie folgt:

`s3_use_arn_region`- Einstellung für gemeinsam genutzte AWS **`config`** Dateien, **`AWS_S3_USE_ARN_REGION`**- Umgebungsvariable, **`aws.s3UseArnRegion`**- JVM-Systemeigenschaft: nur Java/Kotlin , Um den Wert direkt im Code zu konfigurieren, wenden Sie sich direkt an Ihr spezielles SDK.

Diese Einstellung steuert, ob das SDK den Access Point-ARN verwendet AWS-Region , um den regionalen Endpunkt für die Anfrage zu erstellen. Das SDK überprüft, ob der ARN von derselben AWS Partition bereitgestellt AWS-Region wird, auf der der Client konfiguriert ist, AWS-Region um partitionsübergreifende Aufrufe zu verhindern, die höchstwahrscheinlich fehlschlagen. Wenn mehrfach definiert, hat die vom Code konfigurierte Einstellung Vorrang, gefolgt von der Einstellung der Umgebungsvariablen.

Standardwert: `false`

Zulässige Werte:

- **`true`**— Das SDK verwendet AWS-Region beim Erstellen des Endpunkts die ARNs anstelle der vom Client konfigurierten. AWS-Region Ausnahme: Wenn es sich bei der Konfiguration des Clients um ein FIPS AWS-Region handelt AWS-Region, muss es mit den ARNs übereinstimmen. AWS-Region Andernfalls wird ein Fehler ausgegeben.
- **`false`**— Das SDK verwendet AWS-Region bei der Erstellung des Endpunkts die Konfiguration des Clients.

Support durch AWS SDKs und Tools

Die folgenden SDKs unterstützen die in diesem Thema beschriebenen Funktionen und Einstellungen. Alle teilweisen Ausnahmen werden vermerkt. Alle Einstellungen für JVM-Systemeigenschaften werden AWS SDK für Kotlin nur von AWS SDK für Java und vom unterstützt.

SDK	Unterstützt	Hinweise oder weitere Informationen
AWS CLI v2	Ja	
SDK for C++	Ja	
SDK for Go V2 (1.x)	Ja	

SDK	Unterstützt	Hinweise oder weitere Informationen
SDK for Go 1.x (V1)	Ja	Um die Einstellungen für gemeinsam genutzte `config` Dateien zu verwenden, müssen Sie das Laden aus der Konfigurationsdatei aktivieren; siehe Sessions.
SDK for Java 2.x	Ja	
SDK for Java 1.x	Ja	Die JVM-Systemeigenschaft wird nicht unterstützt.
SDK für 3.x JavaScript	Ja	
SDK für 2.x JavaScript	Ja	
SDK für Kotlin	Ja	
SDK for .NET 4.x	Ja	
SDK for .NET 3.x	Ja	Entspricht nicht der Standardpriorität; der Wert einer gemeinsam genutzten `config` Datei hat Vorrang vor der Umgebungsvariablen.
SDK for PHP 3.x	Ja	
SDK for Python (Boto3)	Ja	
SDK for Ruby 3.x	Ja	
SDK für Rust	Nein	
SDK für Swift	Nein	
Tools für PowerShell V5	Ja	Entspricht nicht der Standardpriorität; der Wert einer gemeinsam genutzten `config` Datei hat Vorrang vor der Umgebungsvariablen.
Tools für V4 PowerShell	Ja	Entspricht nicht der Standardpriorität; der Wert einer gemeinsam genutzten `config` Datei hat Vorrang vor der Umgebungsvariablen.

Amazon S3 Multi-Region S3-Zugriffspunkte

> **Note**
>
> Hilfe zum Verständnis des Layouts von Einstellungsseiten oder zur Interpretation der nachfolgenden Tabelle Support by AWS SDKs and Tools finden Sie unterDie Einstellungsseiten dieses Handbuchs verstehen.

Amazon S3 Multi-Region Access Points bieten einen globalen Endpunkt, über den Anwendungen Anfragen von Amazon S3 S3-Buckets bearbeiten können, die sich in mehreren AWS-Regionen befinden. Sie können Multi-Region Access Points verwenden, um Anwendungen mit mehreren Regionen mit derselben Architektur zu erstellen, die in einer einzelnen Region verwendet wird, und diese Anwendungen dann überall auf der Welt ausführen.

Weitere Informationen zu Multi-Region Access Points finden Sie unter Multi-Region Access Points in Amazon S3 im Amazon S3 S3-Benutzerhandbuch.

Weitere Informationen zu Multi-Region Access Point Amazon Resource Names (ARNs) finden Sie unter Anfragen über einen Multi-Region Access Point stellen im Amazon S3 S3-Benutzerhandbuch.

Weitere Informationen zum Erstellen von Multi-Region Access Points finden Sie unter Managing Multi-Region Access Points im Amazon S3 S3-Benutzerhandbuch.

Der SigV4A-Algorithmus ist die Signierungsimplementierung, die zum Signieren der globalen Regionsanfragen verwendet wird. Dieser Algorithmus wird vom SDK durch eine Abhängigkeit von der abgerufen. AWS Common Runtime (CRT) -Bibliotheken

Konfigurieren Sie diese Funktionalität wie folgt:

`s3_disable_multiregion_access_points`- Einstellung für gemeinsam genutzte AWS **`config`** Dateien, **`AWS_S3_DISABLE_MULTIREGION_ACCESS_POINTS`**- Umgebungsvariable, **`aws.s3DisableMultiRegionAccessPoints`**- JVM-Systemeigenschaft: nur Java/Kotlin , Um den Wert direkt im Code zu konfigurieren, wenden Sie sich direkt an Ihr spezielles SDK.

Diese Einstellung steuert, ob das SDK möglicherweise regionsübergreifende Anfragen versucht. Wenn mehrfach definiert, hat die vom Code konfigurierte Einstellung Vorrang, gefolgt von der Einstellung der Umgebungsvariablen.

Standardwert: `false`

Zulässige Werte:

- **`true`**— Stoppt die Verwendung von regionsübergreifenden Anfragen.
- **`false`**— Ermöglicht regionsübergreifende Anfragen mithilfe von Multi-Region Access Points.

Support durch AWS SDKs und Tools

Die folgenden SDKs unterstützen die in diesem Thema beschriebenen Funktionen und Einstellungen. Alle teilweisen Ausnahmen werden vermerkt. Alle Einstellungen für JVM-Systemeigenschaften werden AWS SDK für Kotlin nur von AWS SDK für Java und vom unterstützt.

SDK	Unterstützt	Hinweise oder weitere Informationen
AWS CLI v2	Ja	
SDK for C++	Ja	
SDK for Go V2 (1.x)	Ja	
SDK for Go 1.x (V1)	Nein	
SDK for Java 2.x	Ja	
SDK for Java 1.x	Nein	
SDK für 3.x JavaScript	Ja	
SDK für 2.x JavaScript	Nein	
SDK für Kotlin	Ja	
SDK for .NET 4.x	Ja	
SDK for .NET 3.x	Ja	
SDK for PHP 3.x	Ja	
SDK for Python (Boto3)	Ja	
SDK for Ruby 3.x	Ja	

SDK	U zt	Hinweise oder weitere Informationen
SDK für Rust	Ja	
SDK für Swift	Nein	
Tools für PowerShell V5	Ja	
Tools für V4 PowerShell	Ja	

S3 Express One Zone-Sitzungsauthentifizierung

> **Note**
>
> Hilfe zum Verständnis des Layouts von Einstellungsseiten oder zur Interpretation der nachfolgenden Tabelle Support by AWS SDKs and Tools finden Sie unterDie Einstellungsseiten dieses Handbuchs verstehen.

S3 Express One Zone ist die Hochleistungsspeicherklasse von Amazon S3, die eine Latenz im einstelligen Millisekundenbereich für häufig abgerufene Daten bietet. Wenn Sie S3 Express One Zone-Buckets verwenden, verwenden AWS SDKs und Tools automatisch eine sitzungsbasierte Authentifizierung, die für die Autorisierung von Datenanfragen mit geringer Latenz optimiert ist. Sie verwenden Sitzungstoken mit zonalen Vorgängen (auf Objektebene), um die mit der Autorisierung verbundene Latenz auf eine Reihe von Anfragen in einer Sitzung zu verteilen, wodurch der Authentifizierungsaufwand reduziert und die allgemeine Anforderungsleistung verbessert wird.

S3 Express One Zone-Buckets verwenden ein bestimmtes Benennungsformat, das die Availability Zone-ID enthält, z. B. `bucket-name--usw2-az1--x-s3` Wenn das SDK dieses Benennungsmuster erkennt, leitet es Anfragen automatisch an die entsprechenden S3 Express One Zone-Endpunkte weiter und wendet den optimierten Authentifizierungsablauf an. Bei der Sitzungsauthentifizierung werden temporäre, bucket-spezifische Anmeldeinformationen erstellt, die einen Zugriff auf Ihren Bucket mit geringer Latenz ermöglichen und vom SDK automatisch zwischengespeichert und aktualisiert werden. Weitere Informationen finden Sie unter S3 Express One Zone im Amazon S3-Benutzerhandbuch.

Standardmäßig ist die Sitzungsauthentifizierung für S3 Express One Zone-Buckets aktiviert.

Konfigurieren Sie diese Funktionalität wie folgt:

`s3_disable_express_session_auth`- Einstellung für gemeinsam genutzte AWS **`config`** Dateien, **`AWS_S3_DISABLE_EXPRESS_SESSION_AUTH`**- Umgebungsvariable, **`aws.disableS3ExpressAuth`**- JVM-Systemeigenschaft: nur Java/Kotlin

Steuert, ob die S3 Express One Zone-Sitzungsauthentifizierung deaktiviert ist. Wenn auf gesetzt`true`, verwendet das SDK die Standard-SigV4-Authentifizierung für S3 Express One Zone-Buckets anstelle der Sitzungsauthentifizierung.

Standardwert: `false`

Zulässige Werte:

- **`true`**— Deaktivieren Sie die S3 Express One Zone-Sitzungsauthentifizierung.
- **`false`**— Aktivieren Sie die S3 Express One Zone-Sitzungsauthentifizierung.

Beispiel für die Einstellung dieses Werts in der `config` Datei:

```
[default]
s3_disable_express_session_auth=true
```

Linux/macOS Beispiel für das Setzen von Umgebungsvariablen über die Befehlszeile:

```
export AWS_S3_DISABLE_EXPRESS_SESSION_AUTH=true
```

Windows-Beispiel für das Setzen von Umgebungsvariablen über die Befehlszeile:

```
setx AWS_S3_DISABLE_EXPRESS_SESSION_AUTH true
```

Support durch AWS SDKs und Tools

Die folgenden SDKs unterstützen die in diesem Thema beschriebenen Funktionen und Einstellungen. Alle teilweisen Ausnahmen werden vermerkt. Alle Einstellungen für JVM-Systemeigenschaften werden AWS SDK für Kotlin nur von AWS SDK für Java und vom unterstützt.

SDK	Unterzt	Hinweise oder weitere Informationen
AWS CLI v2	Ja	
AWS CLI v1	Ja	
SDK for C++	Ja	
SDK for Go V2 (1.x)	Ja	
SDK for Go 1.x (V1)	Nein	Um die Einstellungen für gemeinsam genutzte `config` Dateien zu verwenden, müssen Sie das Laden aus der Konfigurationsdatei aktivieren. Weitere Informationen finden Sie unter Sessions.
SDK for Java 2.x	Ja	
SDK for Java 1.x	Nein	
SDK für 3.x JavaScript	Ja	
SDK für 2.x JavaScript	Nein	
SDK für Kotlin	Ja	Die JVM-Systemeigenschaft ist. `aws.s3DisableExpressSessionAuth`
SDK for .NET 4.x	Ja	
SDK for .NET 3.x	Ja	
SDK for PHP 3.x	Ja	
SDK for Python (Boto3)	Ja	
SDK for Ruby 3.x	Ja	
SDK für Rust	Ja	
SDK für Swift	Ja	

SDK	Unterzt	Hinweise oder weitere Informationen
Tools für PowerShell V5	Ja	
Tools für V4 PowerShell	Ja	

Authentifizierungsschema

> **Note**
>
> Hilfe zum Verständnis des Layouts von Einstellungsseiten oder zur Interpretation der nachfolgenden Tabelle Support by AWS SDKs and Tools finden Sie unterDie Einstellungsseiten dieses Handbuchs verstehen.

AWS Dienste unterstützen mehrere Authentifizierungsschemata, z. B. AWS Signature Version 4 (Sigv4) und AWS Signature Version 4a (SigV4a). Standardmäßig wählen SDKs Authentifizierungsschemas auf der Grundlage von Servicemodelldefinitionen aus und priorisieren Schemata, die die beste Kompatibilität bieten. Sie können jedoch Ihr bevorzugtes Authentifizierungsschema konfigurieren, um es für bestimmte Anforderungen zu optimieren.

Im Gegensatz zu Sigv4 sind mit SigV4a signierte Anfragen mehrfach gültig. AWS-Regionen SigV4a bietet eine verbesserte Verfügbarkeit durch regionsübergreifende Anforderungssignierung, wodurch bei regionalen Störungen ein automatisches Failover auf Backup-Regionen ermöglicht wird. Dies ist besonders vorteilhaft für globale Dienste wie AWS Identity and Access Management Amazon CloudFront.

Weitere Informationen zu diesen beiden Authentifizierungsschemata finden Sie im IAM-Benutzerhandbuch unter AWS Signature Version 4 für API-Anfragen.

Konfigurieren Sie diese Funktionalität wie folgt:

`auth_scheme_preference`- Einstellung für gemeinsam genutzte AWS **`config`** Dateien, **`AWS_AUTH_SCHEME_PREFERENCE`**- Umgebungsvariable, **`aws.authSchemePreference`**- JVM-Systemeigenschaft: nur Java/Kotlin

Gibt eine durch Kommas getrennte Liste bevorzugter Authentifizierungsschemas in der Reihenfolge ihrer Priorität an. Wenn ein Dienst mehrere Authentifizierungsschemata unterstützt,

versucht das SDK, Schemas aus dieser Liste in der angegebenen Reihenfolge zu verwenden, und greift auf das Standardverhalten zurück, wenn keines der bevorzugten Schemas verfügbar ist.

Standardwert: Keiner.

Gültige Werte: Eine durch Kommas getrennte Liste mit einem oder mehreren der folgenden Werte:

- **`sigv4`**— Signature Version 4 (schnellste Leistung, nur eine Region)
- **`sigv4a`**— Signaturversion 4a (verbesserte Verfügbarkeit, regionsübergreifende Unterstützung, hat eine langsamere Signaturleistung als Sigv4)
- **`httpBearerAuth`**— Authentifizierung mit HTTP-Bearer-Tokens

Leerzeichen und Tabulatorzeichen zwischen Schemanamen werden ignoriert.

Beispiel für die Einstellung dieses Werts in der `config` Datei, sodass SigV4a bevorzugt wird:

```
[default]
auth_scheme_preference=sigv4a,sigv4
```

`sigv4a_signing_region_set`- Einstellung für gemeinsam genutzte Dateien AWS **`config`**, **`AWS_SIGV4A_SIGNING_REGION_SET`**- Umgebungsvariable

Gibt eine durch Kommas getrennte Liste von AWS-Regionen Sigv4A-Signaturen für mehrere Regionen an. Dies wird als Standardregion für die Anforderung verwendet, wenn SigV4A das gewählte Authentifizierungsschema ist.

Standardwert: Wird durch die Anfrage bestimmt.

Gültige Werte: Comma-separated Liste von AWS-Regionen. Leerzeichen und Tabulatorzeichen zwischen Regionen werden ignoriert.

Support durch AWS SDKs und Tools

Die folgenden SDKs unterstützen die in diesem Thema beschriebenen Funktionen und Einstellungen. Alle teilweisen Ausnahmen werden vermerkt. Alle Einstellungen für JVM-Systemeigenschaften werden AWS SDK für Kotlin nur von AWS SDK für Java und vom unterstützt.

SDK	Unterstützt	Hinweise oder weitere Informationen
AWS CLI v2	Ja	
SDK for C++	Ja	
SDK for Go V2 (1.x)	Ja	
SDK for Go 1.x (V1)	Nein	
SDK for Java 2.x	Ja	
SDK for Java 1.x	Nein	
SDK für 3.x JavaScript	Ja	
SDK für 2.x JavaScript	Nein	
SDK für Kotlin	Ja	
SDK for .NET 4.x	Ja	
SDK for .NET 3.x	Nein	
SDK for PHP 3.x	Ja	
SDK for Python (Boto3)	Ja	
SDK for Ruby 3.x	Ja	
SDK für Rust	Ja	
SDK für Swift	Ja	
Tools für PowerShell V5	Ja	
Tools für V4 PowerShell	Nein	

AWS-Region

> **Note**
>
> Hilfe zum Verständnis des Layouts von Einstellungsseiten oder zur Interpretation der nachfolgenden Tabelle Support by AWS SDKs and Tools finden Sie unterDie Einstellungsseiten dieses Handbuchs verstehen.

AWS-Regionen sind ein wichtiges Konzept, das Sie verstehen sollten, wenn Sie damit AWS-Services arbeiten.

Mit AWS-Regionen können Sie auf diejenigen zugreifen AWS-Services , die sich physisch in einem bestimmten geografischen Gebiet befinden. Dies kann nützlich sein, damit Ihre Daten und Anwendungen in der Nähe ausgeführt werden, wo Sie und Ihre Benutzer darauf zugreifen. Regionen bieten Fehlertoleranz, Stabilität und Ausfallsicherheit und können auch die Latenz verkürzen. Mit Regionen können Sie redundante Ressourcen einrichten, die verfügbar bleiben und von einem regionalen Ausfall nicht betroffen sind.

Die meisten AWS-Service Anfragen beziehen sich auf eine bestimmte geografische Region. Die Ressourcen, die Sie in einer Region erstellen, sind in keiner anderen Region vorhanden, es sei denn, Sie verwenden ausdrücklich eine von einem angebotene Replikationsfunktion AWS-Service. Beispielsweise unterstützen Amazon S3 und Amazon EC2 die regionsübergreifende Replikation. Einige Dienste, z. B. IAM, verfügen nicht über regionale Ressourcen.

Das Allgemeine AWS-Referenzenthält Informationen zu folgenden Themen:

- Informationen zur Beziehung zwischen Regionen und Endpunkten sowie eine Liste der vorhandenen regionalen Endpunkte finden Sie unter AWS Dienstendpunkte.
- Eine aktuelle Liste aller unterstützten Regionen und Endpunkte für die einzelnen Regionen finden Sie unter Dienstendpunkte und AWS-Service Kontingente.

Service-Clients erstellen

Für den programmgesteuerten Zugriff AWS-Services verwenden SDKs jeweils einen Client class/object . AWS-Service Wenn Ihre Anwendung beispielsweise auf Amazon EC2 zugreifen muss, würde Ihre Anwendung ein Amazon EC2 EC2-Client-Objekt als Schnittstelle zu diesem Service erstellen.

Wenn im Code selbst keine Region explizit für den Client angegeben ist, verwendet der Client standardmäßig die Region, die in der folgenden Einstellung festgelegt ist. `region` Die aktive Region für einen Client kann jedoch explizit für jedes einzelne Client-Objekt festgelegt werden. Die Einstellung der Region auf diese Weise hat Vorrang vor allen globalen Einstellungen für diesen bestimmten Service-Client. Die alternative Region wird bei der Instanziierung dieses Clients spezifisch für Ihr SDK angegeben (lesen Sie in Ihrem spezifischen SDK-Handbuch oder in der Codebasis Ihres SDK nach).

Konfigurieren Sie diese Funktionalität wie folgt:

`region`- Einstellung für gemeinsam genutzte AWS **`config`** Dateien, **`AWS_REGION`**-Umgebungsvariable, **`aws.region`**- JVM-Systemeigenschaft: nur Java/Kotlin

Gibt den Standard AWS-Region an, der für AWS Anfragen verwendet werden soll. Diese Region wird für SDK-Dienstanforderungen verwendet, für die keine bestimmte Region zur Verwendung vorgesehen ist.

Standardwert: Keiner. Sie müssen diesen Wert explizit angeben.

Zulässige Werte:

- Alle für den ausgewählten Dienst verfügbaren Regionalcodes, wie sie in der AWS Allgemeinen Referenz unter AWS Dienstendpunkte aufgeführt sind. Der Wert `us-east-1` legt beispielsweise den Endpunkt auf den Osten der AWS-Region USA (Nord-Virginia) fest.
- `aws-global`gibt den globalen Endpunkt für Services an, die zusätzlich zu regionalen Endpunkten auch einen separaten globalen Endpunkt unterstützen, wie AWS -Security-Token-Service (AWS STS) und Amazon Simple Storage Service (Amazon S3).

Beispiel für die Einstellung dieses Werts in der `config` Datei:

```
[default]
region = us-west-2
```

Linux/macOS Beispiel für das Setzen von Umgebungsvariablen über die Befehlszeile:

```
export AWS_REGION=us-west-2
```

Windows-Beispiel für das Setzen von Umgebungsvariablen über die Befehlszeile:

```
setx AWS_REGION us-west-2
```

Die meisten SDKs verfügen über ein „Konfiguration“ -Objekt, mit dem die Standardregion im Anwendungscode festgelegt werden kann. Einzelheiten finden Sie in Ihrem spezifischen AWS SDK-Entwicklerhandbuch.

Support durch AWS SDKs und Tools

Die folgenden SDKs unterstützen die in diesem Thema beschriebenen Funktionen und Einstellungen. Alle teilweisen Ausnahmen werden vermerkt. Alle Einstellungen für JVM-Systemeigenschaften werden AWS SDK für Kotlin nur von AWS SDK für Java und vom unterstützt.

SDK	Unterstützt	Hinweise oder weitere Informationen
AWS CLI v2	Ja	AWS CLI v2 verwendet einen beliebigen Wert in `AWS_REGION` vor einem beliebigen Wert in `AWS_DEFAULT_REGION` (beide Variablen sind geprüft).
AWS CLI v1	Ja	AWS CLI v1 verwendet eine zu diesem `AWS_DEFAULT_REGION` Zweck benannte Umgebungsvariable.
SDK for C++	Ja	
SDK for Go V2 (1.x)	Ja	
SDK for Go 1.x (V1)	Ja	Um die Einstellungen für gemeinsam genutzte `config` Dateien zu verwenden, müssen Sie das Laden aus der Konfigurationsdatei aktivieren. Weitere Informationen finden Sie unter Sessions.
SDK for Java 2.x	Ja	
SDK for Java 1.x	Ja	
SDK für 3.x JavaScript	Ja	
SDK für 2.x JavaScript	Ja	

SDK	Unterstützt	Hinweise oder weitere Informationen
SDK für Kotlin	Ja	
SDK for .NET 4.x	Ja	
SDK for .NET 3.x	Ja	
SDK for PHP 3.x	Ja	
SDK for Python (Boto3)	Ja	Dieses SDK verwendet eine Umgebungsvariable, die zu diesem Zweck benannt ist. `AWS_DEFAULT_REGION`
SDK for Ruby 3.x	Ja	
SDK für Rust	Ja	
SDK für Swift	Ja	
Tools für PowerShell V5	Ja	
Tools für V4 PowerShell	Ja	

AWS STS Regionale -Endpunkte

> **Note**
>
> Hilfe zum Verständnis des Layouts von Einstellungsseiten oder zur Interpretation der nachfolgenden Tabelle Support by AWS SDKs and Tools finden Sie unterDie Einstellungsseiten dieses Handbuchs verstehen.

AWS -Security-Token-Service (AWS STS) ist sowohl als globaler als auch als regionaler Service verfügbar. Einige AWS SDKs und CLIs verwenden standardmäßig den globalen Dienstendpunkt (`https://sts.amazonaws.com`), während andere die regionalen Dienstendpunkte () verwenden. `https://sts.{region_identifier}.{partition_domain}` In Regionen, die standardmäßig aktiviert sind, werden Anfragen an den AWS STS globalen Endpunkt automatisch in derselben Region bearbeitet, aus der die Anfrage stammt. In Opt-in-Regionen werden Anfragen an den AWS

STS globalen Endpunkt von einer einzigen Person AWS-Region, USA Ost (Nord-Virginia), bedient. Weitere Informationen zu AWS STS Endpunkten finden Sie in der AWS -Security-Token-Service API-Referenz unter Endgeräte oder AWS STS im Benutzerhandbuch unter Verwalten. AWS-RegionAWS Identity and Access Management

Es hat sich AWS bewährt, wann immer möglich regionale Endpunkte zu verwenden und Ihre zu konfigurieren. AWS-Region Kunden in anderen Partitionen als kommerziellen Partitionen müssen regionale Endpunkte verwenden. Nicht alle SDKs und Tools unterstützen diese Einstellung, aber alle haben ein definiertes Verhalten in Bezug auf globale und regionale Endpunkte. Weitere Informationen finden Sie im folgenden Abschnitt .

> **Note**
>
> AWS hat in Regionen, die AWS -Security-Token-Service standardmäßig aktiviert sind, Änderungen am globalen Endpunkt (`https://sts.amazonaws.com`) () vorgenommen, um dessen Stabilität und Leistung zu verbessern.AWS STS AWS STS Anfragen an den globalen Endpunkt werden automatisch in denselben Workloads bearbeitet AWS-Region wie Ihre Workloads. Diese Änderungen werden nicht in Opt-in-Regionen bereitgestellt. Wir empfehlen, dass Sie die entsprechenden AWS STS regionalen Endpunkte verwenden. Weitere Informationen finden Sie im AWS Identity and Access Management Benutzerhandbuch unter Änderungen an AWS STS globalen Endpunkten.

Für SDKs und Tools, die diese Einstellung unterstützen, können Kunden die Funktionalität wie folgt konfigurieren:

`sts_regional_endpoints`- Einstellung für gemeinsam genutzte AWS **`config`** Dateien,
`AWS_STS_REGIONAL_ENDPOINTS`- Umgebungsvariable

Diese Einstellung gibt an, wie das SDK oder Tool den AWS-Service Endpunkt bestimmt, über den es mit dem AWS -Security-Token-Service (AWS STS) kommuniziert.

Standardwert:`regional`, siehe Ausnahmen in der folgenden Tabelle.

> **Note**
>
> Alle neuen SDK-Hauptversionen, die nach Juli 2022 veröffentlicht werden, werden standardmäßig auf`regional`. Neue SDK-Hauptversionen könnten diese Einstellung und dieses `regional` Nutzungsverhalten entfernen. Um future Auswirkungen dieser

Änderung zu verringern, empfehlen wir Ihnen, nach Möglichkeit mit `regional` der Verwendung in Ihrer Anwendung zu beginnen.

Gültige Werte: (Empfohlener Wert:*`regional`*)

- **`legacy`**— Verwendet den globalen AWS STS Endpunkt,`sts.amazonaws.com`.
- **`regional`**— Das SDK oder Tool verwendet immer den AWS STS Endpunkt für die aktuell konfigurierte Region. Wenn der Client beispielsweise für die Verwendung konfiguriert ist`us-west-2`, AWS STS werden alle Aufrufe an den regionalen Endpunkt `sts.us-west-2.amazonaws.com` statt an den globalen `sts.amazonaws.com` Endpunkt getätigt. Um eine Anforderung an den globalen Endpunkt zu senden, während diese Einstellung aktiviert ist, können Sie die Region auf `aws-global` festlegen.

Beispiel für das Einstellen dieser Werte in der `config` Datei:

```
[default]
sts_regional_endpoints = regional
```

Linux/macOS Beispiel für das Setzen von Umgebungsvariablen über die Befehlszeile:

```
export AWS_STS_REGIONAL_ENDPOINTS=regional
```

Windows-Beispiel für das Setzen von Umgebungsvariablen über die Befehlszeile:

```
setx AWS_STS_REGIONAL_ENDPOINTS regional
```

Support durch AWS SDKs und Tools

> **Note**
>
> Es hat sich AWS bewährt, wann immer möglich regionale Endpunkte zu verwenden und Ihre AWS-Region zu konfigurieren.

In der folgenden Tabelle finden Sie eine Zusammenfassung für Ihr SDK oder Tool:

- Unterstützt die Einstellung: Gibt an, ob die gemeinsam genutzte `config` Dateivariable und die Umgebungsvariable für regionale STS-Endpunkte unterstützt werden.
- Standardeinstellungswert: Der Standardwert der Einstellung, sofern er unterstützt wird.
- Standard-Ziel-STS-Endpunkt des Service-Clients: Welcher Standardendpunkt wird vom Client verwendet, auch wenn die Einstellung zur Änderung nicht verfügbar ist.
- Fallback-Verhalten des Service-Clients: Was tut das SDK, wenn es einen regionalen Endpunkt verwenden soll, aber keine Region konfiguriert wurde. Dieses Verhalten gilt unabhängig davon, ob ein regionaler Endpunkt aufgrund einer Standardeinstellung verwendet wird oder weil er in der Einstellung ausgewählt `regional` wurde.

In der Tabelle werden auch die folgenden Werte verwendet:

- Globaler Endpunkt:`https://sts.amazonaws.com`.
- Regionaler Endpunkt: Basierend auf der von Ihrer Anwendung AWS-Region verwendeten Konfiguration.
- **`us-east-1`**(Regional): Verwendet den `us-east-1` Regions-Endpunkt, verwendet jedoch längere Sitzungstoken als typische globale Anfragen.

SDK		Standardeinstellungswert	Standard-Ziel-STS-Endpunkt des Service-Clients	Fallback-Verhalten des Serviceclients	Hinweise oder weitere Informationen
AWS CLI v2	Ne	N/A	Regionaler Endpunkt	Globaler Endpunkt	
AWS CLI v1	Ja	`legacy`	Globaler Endpunkt	Globaler Endpunkt	
SDK for C++	Ne	N/A	Regionaler Endpunkt	`us-east-1` (Regional)	
SDK for Go V2 (1.x)	Ne	N/A	Regionaler Endpunkt	Fehler bei der Anfrage	

SDK		Standardeinstellungswert	Standard-Ziel-STS-Endpunkt des Service-Clients	Fallback-Verhalten des Serviceclients	Hinweise oder weitere Informationen
SDK for Go 1.x (V1)	Ja	`legacy`	Globaler Endpunkt	Globaler Endpunkt	Um die Einstellungen für gemeinsam genutzte `config` Dateien zu verwenden, müssen Sie das Laden aus der Konfigurationsdatei aktivieren. Weitere Informationen finden Sie unter Sitzungen.
SDK for Java 2.x	Ne	N/A	Regionaler Endpunkt	Fehler bei der Anfrage	Wenn keine Region konfiguriert ist, verwendet der `AssumeRole` und `AssumeRoleWithWebIdentity` den globalen STS-Endpunkt.
SDK for Java 1.x	Ja	`legacy`	Globaler Endpunkt	Globaler Endpunkt	
SDK für JavaScript 3.x	Ne	N/A	Regionaler Endpunkt	`us-east-1` (Regional)	
SDK für JavaScript 2.x	Ja	`legacy`	Globaler Endpunkt	Globaler Endpunkt	
SDK für Kotlin	Ne	N/A	Regionaler Endpunkt	Globaler Endpunkt	
SDK for .NET 4.x	Ne	N/A	Regionaler Endpunkt	`us-east-1` (Regional)	

SDK		Standardeinstellungswert	Standard-Ziel-STS-Endpunkt des Service-Clients	Fallback-Verhalten des Serviceclients	Hinweise oder weitere Informationen
SDK for .NET 3.x	Ja	`regional`	Globaler Endpunkt	Globaler Endpunkt	
SDK for PHP 3.x	Ja	`regional`	Globaler Endpunkt	Fehler bei der Anfrage	
SDK for Python (Boto3)	Ja	`regional`	Globaler Endpunkt	Globaler Endpunkt	
SDK for Ruby 3.x	Ja	`regional`	Regionaler Endpunkt	Anfrage ist fehlgeschlagen	
SDK für Rust	Ne	N/A	Regionaler Endpunkt	Fehler bei der Anfrage	
SDK für Swift	Ne	N/A	Regionaler Endpunkt	Fehler bei der Anfrage	
Tools für PowerShell V5	Ja	`regional`	Globaler Endpunkt	Globaler Endpunkt	
Tools für PowerShell V4	Ja	`regional`	Globaler Endpunkt	Globaler Endpunkt	

Schutz der Datenintegrität für Amazon S3

> **Note**
>
> Hilfe zum Verständnis des Layouts von Einstellungsseiten oder zur Interpretation der nachfolgenden Tabelle Support by AWS SDKs and Tools finden Sie unterDie Einstellungsseiten dieses Handbuchs verstehen.

Seit einiger Zeit unterstützen AWS SDKs Datenintegritätsprüfungen beim Hoch- oder Herunterladen von Daten aus Amazon Simple Storage Service. Bisher waren diese Prüfungen freiwillig. Jetzt haben wir diese Prüfungen standardmäßig aktiviert und dabei CRC-based Algorithmen wie CRC32 oder CRC64NVME verwendet. Obwohl jedes SDK oder Tool über einen Standardalgorithmus verfügt, können Sie einen anderen Algorithmus wählen. Sie können auch weiterhin manuell eine vorberechnete Prüfsumme für Uploads angeben, wenn Sie möchten. Ein konsistentes Verhalten bei Uploads, mehrteiligen Uploads, Downloads und Verschlüsselungsmodi vereinfacht die clientseitigen Integritätsprüfungen.

Die neuesten Versionen unserer AWS SDKs und berechnen AWS CLI automatisch eine auf der zyklischen Redundanzprüfung (CRC) basierende Prüfsumme für jeden Upload und senden sie an Amazon S3. Amazon S3 berechnet unabhängig eine Prüfsumme auf der Serverseite und validiert sie anhand des angegebenen Werts, bevor das Objekt und seine Prüfsumme dauerhaft in den Metadaten des Objekts gespeichert werden. Durch das Speichern der Prüfsumme in den Metadaten neben dem Objekt kann beim Herunterladen des Objekts dieselbe Prüfsumme automatisch zurückgegeben und auch zur Validierung von Downloads verwendet werden. Sie können die in den Metadaten des Objekts gespeicherte Prüfsumme auch jederzeit überprüfen.

Weitere Informationen über Prüfsummenoperationen, mehrteilige Uploads oder die Liste der unterstützten Prüfsummenalgorithmen finden Sie unter Prüfen der Objektintegrität in Amazon S3 im Amazon Simple Storage Service-Benutzerhandbuch.

Mehrteilige Uploads:

Amazon S3 bietet Entwicklern außerdem konsistente vollständige Objektprüfsummen für einteilige und mehrteilige Uploads.

Beim Hochladen von Dateien in mehreren Teilen berechnen die SDKs Prüfsummen für jeden Teil. Amazon S3 verwendet diese Prüfsummen, um die Integrität jedes Teils über die `UploadPart` API

zu überprüfen. Darüber hinaus überprüft Amazon S3 die Größe und Prüfsumme der gesamten Datei, wenn Sie die `CompleteMultipartUpload` API aufrufen.

Wenn Ihr SDK über einen Amazon S3 Transfer Manager verfügt, der Sie bei mehrteiligen Uploads unterstützt, werden die Prüfsummen für die Teile anhand des in der Tabelle SDK-specific angegebenen Standardalgorithmus validiert. Support durch AWS SDKs und Tools Sie können sich für eine vollständige Objektprüfsumme entscheiden, indem Sie die Einstellung auf `FULL_OBJECT` oder die Verwendung des `checksum_type` CRC64NVME-Algorithmus festlegen.

Wenn Sie eine ältere Version des SDK verwenden oder: AWS CLI

Wenn Ihre Anwendung eine Version des SDK oder Tools vor Dezember 2024 verwendet, berechnet Amazon S3 dennoch eine CRC64NVME-Prüfsumme für neue Objekte und speichert sie in den Objektmetadaten, damit Sie später darauf future können. Sie können den gespeicherten CRC später mit einem CRC vergleichen, der auf Ihrer Seite berechnet wurde, und überprüfen, ob die Netzwerkübertragung korrekt war. Außerdem können Sie den Integritätsschutz immer noch manuell erweitern, indem Sie Ihre eigenen vorberechneten Prüfsummen mit Ihren `PutObjectUploadPart`Oder-Anfragen angeben. Dies ist die Standardmethode, um dieses Problem in älteren Versionen zu beheben.

Konfigurieren Sie diese Funktionalität wie folgt:

`request_checksum_calculation`- Einstellung für gemeinsam genutzte AWS **`config`** Dateien, **`AWS_REQUEST_CHECKSUM_CALCULATION`**- Umgebungsvariable, **`aws.requestChecksumCalculation`**- JVM-Systemeigenschaft: nur Java/Kotlin

Standardmäßig sind Benutzer bereit, beim Senden einer Anfrage eine Prüfsumme für Anfragen zu berechnen. Der Benutzer kann bei der Erstellung der Anfrage jeden der verfügbaren Prüfsummenalgorithmen wählen. Andernfalls wird ein SDK-specific Standardalgorithmus verwendet. In der Support durch AWS SDKs und Tools Tabelle finden Sie den Standardalgorithmus für jedes SDK oder Tool.

Standardwert: `WHEN_SUPPORTED`

Zulässige Werte:

- **`WHEN_SUPPORTED`**— Die Prüfsummenvalidierung wird für alle Anforderungsnutzlasten durchgeführt, sofern sie vom API-Vorgang unterstützt werden, z. B. Datenübertragungen an Amazon S3.

- **`WHEN_REQUIRED`**— Die Prüfsummenvalidierung wird nur durchgeführt, wenn dies für den API-Vorgang erforderlich ist.

`response_checksum_validation`- Einstellung für gemeinsam genutzte AWS **`config`** Dateien, **`AWS_RESPONSE_CHECKSUM_VALIDATION`**- Umgebungsvariable, **`aws.responseChecksumValidation`**- JVM-Systemeigenschaft: nur Java/Kotlin

Standardmäßig haben Benutzer beim Senden einer Anfrage die Überprüfung der Antwortprüfsumme aktiviert. Eine Prüfsumme wird für die Antwort-Nutzlast berechnet und mit dem Checksum-Antwort-Header verglichen. Wenn die Prüfsummenvalidierung fehlschlägt, wird dem Benutzer beim Lesen der Payload ein Fehler gemeldet.

Der Checksum-Antwort-Header gibt auch den Algorithmus für die Prüfsumme an. Der Amazon S3 S3-Client versucht, Antwortprüfsummen für alle Amazon S3 S3-API-Operationen zu validieren, die Prüfsummen unterstützen. Wenn das SDK den angegebenen Prüfsummenalgorithmus jedoch nicht implementiert hat, wird diese Validierung übersprungen.

Standardwert: `WHEN_SUPPORTED`

Zulässige Werte:

- **`WHEN_SUPPORTED`**— Die Prüfsummenvalidierung wird für alle Antwort-Payloads durchgeführt, sofern sie vom API-Vorgang unterstützt werden, z. B. bei Datenübertragungen an Amazon S3.
- **`WHEN_REQUIRED`**— Die Prüfsummenvalidierung wird nur durchgeführt, wenn sie von der API-Operation unterstützt wird und der Aufrufer die Prüfsumme für die Operation explizit aktiviert hat. Zum Beispiel, wenn die Amazon S3 `GetObject` S3-API aufgerufen wird und der `ChecksumMode` Parameter auf aktiviert gesetzt ist.

Support durch AWS SDKs und Tools

Die folgenden SDKs unterstützen die in diesem Thema beschriebenen Funktionen und Einstellungen. Alle teilweisen Ausnahmen werden vermerkt. Alle Einstellungen für JVM-Systemeigenschaften werden AWS SDK für Kotlin nur von AWS SDK für Java und vom unterstützt.

> **Note**
>
> In der folgenden Tabelle bezieht sich „CRT“ auf das AWS Common Runtime (CRT) - Bibliotheken und erfordert möglicherweise das Hinzufügen einer zusätzlichen Abhängigkeit zu Ihrem Projekt.

SDK	Unterzt	Standardmäßiger Prüfsummenalgorithmus	Unterstützte Prüfsummenalgorithmen	Hinweise oder weitere Informationen
AWS CLI v2	Ja	CRC64NVME	CRC64NVME, CRC32, CRC32C, SHA1, SHA256, xxHash3, xxHash64, xxHASH128, SHA512	Für AWS CLI Version 1 sind der Standardalgorithmus und die unterstützten Algorithmen identisch mit Python (Boto3).
SDK for C++	Ja	CRC64NVME	CRC64NVME, CRC32, CRC32C, SHA1, SHA256, xxHash3, xxHash64, xxHASH128, SHA512	
SDK for Go V2 (1.x)	Ja	CRC32	CRC64NVME, CRC32, CRC32C, SHA1, SHA256	
SDK for Go 1.x (V1)	Nein			
SDK for Java 2.x	Ja	CRC32	CRC32, CRC32C, SHA1, SHA256 Nur über CRT: CRC64NVME, xxHash3, xxHash64, xxHash128, SHA512	
SDK for Java 1.x	Nein			
SDK für 3.x JavaScript	Ja	CRC32	CRC32, CRC32C, SHA1, SHA256	

SDK	Unterzt	Standardmäßiger Prüfsummenalgorithmus	Unterstützte Prüfsummenalgorithmen	Hinweise oder weitere Informationen
SDK JavaScript für 2.x	Nein			
SDK für Kotlin	Ja	CRC32	CRC32, CRC32C, SHA1, SHA256	
SDK for .NET 4.x	Ja	CRC32	CRC32, CRC32C, SHA1, SHA256, SHA512	
SDK for .NET 3.x	Ja	CRC32	CRC32, CRC32C, SHA1, SHA256, SHA512	
SDK for PHP 3.x	Ja	CRC32	CRC32, SHA1, SHA256 Nur über CRT: CRC32C	`awscrt`Für die Verwendung von CRC32C ist eine Erweiterung erforderlich.
SDK for Python (Boto3)	Ja	CRC32	CRC32, SHA1, SHA256 Nur über CRT: CRC32C, CRC64NVME, xxHash3, xxHash64, xxHash128, SHA512	

SDK	Unterzt	Standardmäßiger Prüfsummenalgorithmus	Unterstützte Prüfsummenalgorithmen	Hinweise oder weitere Informationen
SDK for Ruby 3.x	Ja	CRC32	CRC32, SHA1, SHA256 Nur über CRT: CRC64NVME, CRC32C	
SDK für Rust	Ja	CRC32	CRC64NVME, CRC32, CRC32C, SHA1, SHA256	
SDK für Swift	Ja	CRC32	CRC64NVME, CRC32, CRC32C, SHA1, SHA256	CRT-Abhängigkeit ist für alle Algorithmen erforderlich.
Tools für V5 PowerShell	Ja	CRC32	CRC32, CRC32C, SHA1, SHA256, xxHASH3, SHA512	
Tools PowerShell für V4	Ja	CRC32	CRC32, CRC32C, SHA1, SHA256, SHA512	

Dual-stack und FIPS-Endpunkte

> **Note**
>
> Hilfe zum Verständnis des Layouts von Einstellungsseiten oder zur Interpretation der nachfolgenden Tabelle Support by AWS SDKs and Tools finden Sie unterDie Einstellungsseiten dieses Handbuchs verstehen.

Konfigurieren Sie diese Funktionalität wie folgt:

`use_dualstack_endpoint`- Einstellung für gemeinsam genutzte AWS **`config`** Dateien, **`AWS_USE_DUALSTACK_ENDPOINT`**- Umgebungsvariable, **`aws.useDualstackEndpoint`**- JVM-Systemeigenschaft: nur Java/Kotlin

Schaltet ein oder aus, ob das SDK Anfragen an Dual-Stack-Endpunkte sendet. Weitere Informationen zu Dual-Stack-Endpunkten, die sowohl IPv4- als auch IPv6-Verkehr unterstützen, finden Sie unter Verwenden von Amazon S3 S3-Dual-Stack-Endpunkten im Amazon Simple Storage Service-Benutzerhandbuch. Dual-stackEndpunkte sind für einige Dienste in einigen Regionen verfügbar.

Standardwert: `false`

Zulässige Werte:

- **`true`**— Das SDK oder Tool versucht, Dual-Stack-Endpunkte zu verwenden, um Netzwerkanfragen zu stellen. Wenn für den Dienst kein Dual-Stack-Endpunkt existiert and/or AWS-Region, schlägt die Anfrage fehl.
- **`false`**— Das SDK oder Tool verwendet keine Dual-Stack-Endpunkte, um Netzwerkanfragen zu stellen.

`use_fips_endpoint`- Einstellung für gemeinsam genutzte Dateien AWS **`config`**, **`AWS_USE_FIPS_ENDPOINT`**- Umgebungsvariable, **`aws.useFipsEndpoint`**- JVM-Systemeigenschaft: nur Java/Kotlin

Schaltet ein oder aus, ob das SDK oder das Tool Anfragen an FIPS-compliant Endpunkte sendet. Bei den Federal Information Processing Standards (FIPS) handelt es sich um eine Reihe von Sicherheitsanforderungen der US-Regierung für Daten und deren Verschlüsselung. Regierungsbehörden, Partner und Personen, die mit der Bundesregierung Geschäfte machen möchten, müssen sich an die FIPS-Richtlinien halten. Im Gegensatz zu AWS Standardendpunkten verwenden FIPS-Endpunkte eine TLS-Softwarebibliothek, die anhand von FIPS 140 validiert wurde. Wenn diese Einstellung aktiviert ist und kein FIPS-Endpunkt für den Dienst in Ihrem System vorhanden ist AWS-Region, schlägt der Anruf möglicherweise fehl. AWS Service-specific Endpunkteund die `--endpoint-url` Option zum AWS Command Line Interface Überschreiben dieser Einstellung.

Weitere Informationen zu anderen Methoden zur Angabe von AWS-Region FIPS-Endpunkten finden Sie unter FIPS-Endpunkte nach Dienst. Weitere Informationen zu Amazon Elastic Compute Cloud Service-Endpunkten finden Sie unter Dual-stack (IPv4 und IPv6) -Endpoints in der Amazon EC2 API-Referenz.

Standardwert: `false`

Zulässige Werte:

- **`true`**— Das SDK oder Tool sendet Anfragen an Endpunkte. FIPS-compliant
- **`false`**— Das SDK oder Tool sendet keine Anfragen an FIPS-compliant Endpunkte.

Support durch AWS SDKs und Tools

Die folgenden SDKs unterstützen die in diesem Thema beschriebenen Funktionen und Einstellungen. Alle teilweisen Ausnahmen werden vermerkt. Alle Einstellungen für JVM-Systemeigenschaften werden AWS SDK für Kotlin nur von AWS SDK für Java und vom unterstützt.

SDK	Unterstützt	Hinweise oder weitere Informationen
AWS CLI v2	Ja	
SDK for C++	Ja	
SDK for Go V2 (1.x)	Ja	
SDK for Go 1.x (V1)	Ja	Um die Einstellungen für gemeinsam genutzte `config` Dateien zu verwenden, müssen Sie das Laden aus der Konfigurationsdatei aktivieren. Weitere Informationen finden Sie unter Sessions.
SDK for Java 2.x	Ja	
SDK for Java 1.x	Nein	
SDK für 3.x JavaScript	Ja	
SDK für 2.x JavaScript	Ja	
SDK für Kotlin	Ja	
SDK for .NET 4.x	Ja	
SDK for .NET 3.x	Ja	
SDK for PHP 3.x	Ja	

SDK	Unterstützt	Hinweise oder weitere Informationen
SDK for Python (Boto3)	Ja	
SDK for Ruby 3.x	Ja	
SDK für Rust	Ja	
SDK für Swift	Ja	
Tools für PowerShell V5	Ja	
Tools für V4 PowerShell	Ja	

Endpunkterkennung

> **Note**
>
> Hilfe zum Verständnis des Layouts von Einstellungsseiten oder zur Interpretation der nachfolgenden Tabelle Support by AWS SDKs and Tools finden Sie unterDie Einstellungsseiten dieses Handbuchs verstehen.

SDKs verwenden Endpoint Discovery für den Zugriff auf Dienstendpunkte (URLs für den Zugriff auf verschiedene Ressourcen) und behalten gleichzeitig die Flexibilität, URLs nach AWS Bedarf zu ändern. Auf diese Weise kann Ihr Code automatisch neue Endpunkte erkennen. Für einige Dienste gibt es keine festen Endpunkte. Stattdessen erhalten Sie die verfügbaren Endpunkte während der Laufzeit, indem Sie eine Anfrage stellen, um zuerst die Endpunkte abzurufen. Nach dem Abrufen der verfügbaren Endpunkte verwendet der Code dann den Endpunkt, um auf andere Operationen zuzugreifen. Für Amazon Timestream stellt das SDK beispielsweise eine `DescribeEndpoints` Anfrage zum Abrufen der verfügbaren Endpunkte und verwendet diese Endpunkte dann, um bestimmte Operationen wie oder abzuschließen. `CreateDatabase` `CreateTable`

Konfigurieren Sie diese Funktionalität wie folgt:

`endpoint_discovery_enabled`- Einstellung für gemeinsam genutzte AWS **`config`** Dateien, **`AWS_ENABLE_ENDPOINT_DISCOVERY`**- Umgebungsvariable, **`aws.endpointDiscoveryEnabled`**- JVM-Systemeigenschaft: nur Java/Kotlin , Um den Wert direkt im Code zu konfigurieren, wenden Sie sich direkt an Ihr spezielles SDK.

Aktiviert oder deaktiviert die Endpunkterkennung für DynamoDB.

Endpoint Discovery ist in Timestream erforderlich und in Amazon DynamoDB optional. Diese Einstellung ist standardmäßig entweder `true` oder, `false` je nachdem, ob der Service eine Endpunkterkennung erfordert, voreingestellt. Timestream-Anfragen sind standardmäßig auf `true` und Amazon DynamoDB DynamoDB-Anfragen standardmäßig auf. `false`

Zulässige Werte:

- **`true`**— Das SDK sollte automatisch versuchen, einen Endpunkt für Dienste zu ermitteln, bei denen die Endpunkterkennung optional ist.
- **`false`**— Das SDK sollte nicht automatisch versuchen, einen Endpunkt für Dienste zu finden, bei denen die Endpunkterkennung optional ist.

Support durch AWS SDKs und Tools

Die folgenden SDKs unterstützen die in diesem Thema beschriebenen Funktionen und Einstellungen. Alle teilweisen Ausnahmen werden vermerkt. Alle Einstellungen für JVM-Systemeigenschaften werden AWS SDK für Kotlin nur von AWS SDK für Java und vom unterstützt.

SDK	Unterstützt	Hinweise oder weitere Informationen
AWS CLI v2	Ja	
SDK for C++	Ja	
SDK for Go V2 (1.x)	Ja	
SDK for Go 1.x (V1)	Ja	Um die Einstellungen für gemeinsam genutzte `config` Dateien zu verwenden, müssen Sie das Laden aus der Konfigurationsdatei aktivieren. Weitere Informationen finden Sie unter Sessions.

SDK	Unterstützt	Hinweise oder weitere Informationen
SDK for Java 2.x	Ja	Das SDK for Java 2.x verwendet `AWS_ENDPOINT_DISCOVERY_ENABLED` für die Umgebungsvariable den Namen.
SDK for Java 1.x	Teilweise	Die JVM-Systemeigenschaft wird nicht unterstützt.
SDK für 3.x JavaScript	Ja	
SDK für 2.x JavaScript	Ja	
SDK für Kotlin	Ja	
SDK for .NET 4.x	Ja	
SDK for .NET 3.x	Ja	
SDK for PHP 3.x	Ja	
SDK for Python (Boto3)	Ja	
SDK for Ruby 3.x	Ja	
SDK für Rust	Teilweise	Wird nur für Timestream unterstützt.
SDK für Swift	Nein	
Tools für PowerShell V5	Ja	
Tools für V4 PowerShell	Ja	

Allgemeine Konfigurationseinstellungen

> **Note**
>
> Hilfe zum Verständnis des Layouts von Einstellungsseiten oder zur Interpretation der nachfolgenden Tabelle Support by AWS SDKs and Tools finden Sie unter Die Einstellungsseiten dieses Handbuchs verstehen.

SDKs unterstützen einige allgemeine Einstellungen, mit denen das allgemeine SDK-Verhalten konfiguriert wird.

Konfigurieren Sie diese Funktionalität wie folgt:

`api_versions`- Einstellung für gemeinsam genutzte AWS **`config`** Dateien

Einige AWS Dienste verwenden mehrere API-Versionen, um die Abwärtskompatibilität zu unterstützen. Standardmäßig verwenden SDK und AWS CLI Operationen die neueste verfügbare API-Version. Wenn Sie für Ihre Anfragen eine bestimmte API-Version benötigen möchten, fügen Sie die `api_versions` Einstellung in Ihr Profil ein.

Standardwert: Keiner. (Die neueste API-Version wird vom SDK verwendet.)

Gültige Werte: Dies ist eine verschachtelte Einstellung, auf die eine oder mehrere eingerückte Zeilen folgen, die jeweils einen AWS Dienst und die zu verwendende API-Version angeben. In der Dokumentation zum AWS Dienst finden Sie Informationen darüber, welche API-Versionen verfügbar sind.

Das Beispiel legt eine bestimmte API-Version für zwei AWS Dienste in der `config` Datei fest. Diese API-Versionen werden nur für Befehle verwendet, die unter dem Profil mit diesen Einstellungen ausgeführt werden. Befehle für jeden anderen Dienst verwenden die neueste Version der API dieses Dienstes.

```
api_versions =
    ec2 = 2015-03-01
    cloudfront = 2015-09-017
```

`ca_bundle`- Einstellung für gemeinsam genutzte AWS **`config`** Dateien, **`AWS_CA_BUNDLE`**-Umgebungsvariable

Gibt den Pfad zu einem benutzerdefinierten Zertifikatspaket (einer Datei mit einer `.pem` Erweiterung) an, das beim Herstellen von SSL/TLS Verbindungen verwendet werden soll.

Standardwert: keiner

Gültige Werte: Geben Sie entweder den vollständigen Pfad oder einen Basisdateinamen an. Wenn es einen Basisdateinamen gibt, versucht das System, das Programm in den durch die `PATH` Umgebungsvariable angegebenen Ordnern zu finden.

Beispiel für die Einstellung dieses Werts in der `config` Datei:

```
[default]
ca_bundle = dev/apps/ca-certs/cabundle-2019mar05.pem
```

Aufgrund von Unterschieden in der Art und Weise, wie Betriebssysteme Pfade behandeln und Pfadzeichen maskieren, finden Sie im Folgenden ein Beispiel für die Einstellung dieses Werts in der `config` Datei unter Windows:

```
[default]
ca_bundle = C:\\Users\\username\\.aws\\aws-custom-bundle.pem
```

Linux/macOS Beispiel für das Setzen von Umgebungsvariablen über die Befehlszeile:

```
export AWS_CA_BUNDLE=/dev/apps/ca-certs/cabundle-2019mar05.pem
```

Windows-Beispiel für das Setzen von Umgebungsvariablen über die Befehlszeile:

```
setx AWS_CA_BUNDLE C:\dev\apps\ca-certs\cabundle-2019mar05.pem
```

`output`- Einstellung für gemeinsam genutzte AWS **`config`** Dateien

Gibt an, wie Ergebnisse in den AWS CLI und anderen AWS SDKs und Tools formatiert werden.

Standardwert: `json`

Zulässige Werte:

- **`json`** – Die Ausgabe erfolgt im JSON-Format.
- **`yaml`** –Die Ausgabe erfolgt im YAML-Format.
- **`yaml-stream`** – Die Ausgabe erfolgt im YAML-Format und wird so auch gestreamt. Streaming ermöglicht eine schnellere Handhabung großer Datentypen.
- **`text`** – Die Ausgabe wird als mehrere Zeilen mit tabulatorgetrennten Zeichenfolgenwerten formatiert. Dies kann nützlich sein, um die Ausgabe an einen Textprozessor wie `grep`, `sed` oder `awk` zu übergeben.
- **`table`** – Die Ausgabe erfolgt in Form einer Tabelle mit den Zeichen +|-, um die Zellenrahmen zu bilden. Normalerweise wird die Information in einem benutzerfreundlichen Format wiedergegeben, das viel einfacher zu lesen ist als die anderen, jedoch programmatisch nicht so nützlich ist.

`parameter_validation`- Einstellung für gemeinsam genutzte Dateien AWS **`config`**

Gibt an, ob das SDK oder das Tool versucht, Befehlszeilenparameter zu überprüfen, bevor sie an den AWS Dienstendpunkt gesendet werden.

Standardwert: `true`

Zulässige Werte:

- **`true`** – Der Standardwert. Das SDK oder Tool führt eine clientseitige Überprüfung der Befehlszeilenparameter durch. Auf diese Weise kann das SDK oder Tool überprüfen, ob die Parameter gültig sind, und es werden einige Fehler erkannt. Das SDK oder Tool kann Anfragen ablehnen, die nicht gültig sind, bevor Anfragen an den AWS Dienstendpunkt gesendet werden.
- **`false`**— Das SDK oder Tool validiert Befehlszeilenparameter nicht, bevor sie an den AWS Dienstendpunkt gesendet werden. Der AWS Dienstendpunkt ist dafür verantwortlich, alle Anfragen zu validieren und Anfragen abzulehnen, die nicht gültig sind.

Support durch AWS SDKs und Tools

Die folgenden SDKs unterstützen die in diesem Thema beschriebenen Funktionen und Einstellungen. Alle teilweisen Ausnahmen werden vermerkt. Alle Einstellungen für JVM-Systemeigenschaften werden AWS SDK für Kotlin nur von AWS SDK für Java und vom unterstützt.

SDK	Unterstützt	Hinweise oder weitere Informationen
AWS CLI v2	Teilweise	`api_versions` wird nicht unterstützt.
SDK for C++	Ja	
SDK for Go V2 (1.x)	Teilweise	`api_versions` und wird `parameter_validation` nicht unterstützt.
SDK for Go 1.x (V1)	Teilweise	`api_versions` und wird `parameter_validation` nicht unterstützt. Um die Einstellungen für gemeinsam genutzte `config` Dateien zu verwenden, müssen Sie das Laden aus der Konfigurationsdatei aktivieren. Weitere Informationen finden Sie unter Sitzungen.

SDK	Unterstützt	Hinweise oder weitere Informationen
SDK for Java 2.x	Nein	
SDK for Java 1.x	Nein	
SDK für 3.x JavaScript	Ja	
SDK für 2.x JavaScript	Ja	
SDK für Kotlin	Nein	
SDK for .NET 4.x	Nein	
SDK for .NET 3.x	Nein	
SDK for PHP 3.x	Ja	
SDK for Python (Boto3)	Ja	
SDK for Ruby 3.x	Ja	
SDK für Rust	Nein	
SDK für Swift	Nein	
Tools für PowerShell V5	Nein	
Tools für V4 PowerShell	Nein	

Host-Präfix-Injektion

> **Note**
>
> Hilfe zum Verständnis des Layouts von Einstellungsseiten oder zur Interpretation der nachfolgenden Tabelle Support by AWS SDKs and Tools finden Sie unter Die Einstellungsseiten dieses Handbuchs verstehen.

Bei der Host-Präfix-Injection handelt es sich um eine Funktion, bei der AWS SDKs dem Hostnamen von Dienstendpunkten für bestimmte API-Operationen automatisch ein Präfix voranstellen. Dieses Präfix kann entweder eine statische Zeichenfolge oder ein dynamischer Wert sein, der Daten aus Ihren Anforderungsparametern enthält.

Wenn Sie beispielsweise Amazon Simple Storage Service verwenden, um Aktionen an Amazon S3 S3-Objekten oder -Buckets durchzuführen, ersetzt das SDK Ihren Bucket-Namen und Ihre AWS-Konto Bucket-ID im endgültigen API-Endpunkt.

Dieses Verhalten ist zwar für normale AWS Dienstendpunkte erforderlich, kann jedoch zu Problemen führen, wenn benutzerdefinierte Endpunkte wie VPC-Endpunkte oder lokale Testtools verwendet werden. In diesen Fällen müssen Sie möglicherweise die Hostpräfixinjektion deaktivieren.

Konfigurieren Sie diese Funktionalität wie folgt:

`disable_host_prefix_injection`- Einstellung für gemeinsam genutzte AWS **`config`** Dateien, **`AWS_DISABLE_HOST_PREFIX_INJECTION`**- Umgebungsvariable, **`aws.disableHostPrefixInjection`**- JVM-Systemeigenschaft: nur Java/Kotlin

Diese Einstellung steuert, ob das SDK oder das Tool den Hostnamen des Endpunkts ändert, indem ein Hostpräfix vorangestellt wird, wie es im Client-Objekt oder in der Variablen Ihres SDK definiert ist.

Standardwert: `false`

Zulässige Werte:

- **`true`**— Deaktiviert die Host-Präfix-Injektion. Das SDK ändert den Hostnamen des Endpunkts nicht.
- **`false`**— Aktiviert die Host-Präfix-Injektion. Das SDK stellt das Hostpräfix dem Hostnamen des Endpunkts voran.

Beispiel für die Einstellung dieses Werts in der `config` Datei:

```
[default]
disable_host_prefix_injection = true
```

Linux/macOS Beispiel für das Setzen von Umgebungsvariablen über die Befehlszeile:

```
export AWS_DISABLE_HOST_PREFIX_INJECTION=true
```

Windows-Beispiel für das Setzen von Umgebungsvariablen über die Befehlszeile:

```
setx AWS_DISABLE_HOST_PREFIX_INJECTION true
```

Beispiele für die Injektion von Hostpräfixen

Die folgende Tabelle mit Beispielen zeigt, wie SDKs den endgültigen Endpunkt ändern, wenn die Hostpräfixinjektion aktiviert und deaktiviert ist.

- Hostpräfix: Die Vorlage der Eigenschaftszeichenfolge für das Hostpräfix, die für das Client-Objekt oder die Variable des SDK im Code festgelegt ist.
- Eingaben: Zusätzliche Eingaben, die für das Client-Objekt oder die Variable des SDK im Code festgelegt wurden.
- Client-Endpunkt: Der abgeleitete Endpunkt des Clients.
- Einstellungswert: Gelöster Wert für die vorherige Einstellung.
- Resultierender Endpunkt: Der resultierende Endpunkt, den der SDK-Client für den API-Aufruf verwendet.

Host-Präfix	Eingaben	Client-Endpunkt	Wert einstellen	Resultierender Endpunkt
„Daten“.	{}	"https://service.us-west-2.amazonaws.com"	false	"https://data.service.us-west-2.amazonaws.com"
„{Eimer} - {AccountId}.“	Eimer: „amzn-s3-demo-bucket1",:" 123456789012" AccountId	"https://service.us-west-2.amazonaws.com"	false	"https://amzn-s3-demo-bucket1-123456789012.service.us-west-2.amazonaws.com"
„Daten“.	{}	"https://override.us-west-2	true	"https://override.us-west-2

Host-Präfix	Eingaben	Client-Endpunkt	Wert einstellen	Resultierender Endpunkt
		.amazonaws.com"(als Override-Endpunkt)		.amazonaws.com"

Support durch AWS SDKs und Tools

Die folgenden SDKs unterstützen die in diesem Thema beschriebenen Funktionen und Einstellungen. Alle teilweisen Ausnahmen werden vermerkt. Alle Einstellungen für JVM-Systemeigenschaften werden AWS SDK für Kotlin nur von AWS SDK für Java und vom unterstützt.

SDK	Unterstützt	Hinweise oder weitere Informationen
AWS CLI v2	Ja	
SDK for C++	Nein	Die Einstellung wird nicht unterstützt, kann aber im Code auf dem Client konfiguriert werden mit: `enableHostPrefixInjection`.
SDK for Go V2 (1.x)	Nein	Kann mithilfe von Middleware deaktiviert werden.
SDK for Go 1.x (V1)	Nein	
SDK for Java 2.x	Nein	Die Einstellung wird nicht unterstützt, kann aber im Code auf dem Client konfiguriert werden mit: `SdkAdvancedClientOption.DISABLE_HOST_PREFIX_INJECTION`.
SDK for Java 1.x	Nein	Die Einstellung wird nicht unterstützt, kann aber im Code auf dem Client konfiguriert werden mit: `withDisableHostPrefixInjection`.
SDK für JavaScript 3.x	Nein	Die Einstellung wird nicht unterstützt, kann aber im Code auf dem Client konfiguriert werden, indem Sie Folgendes verwenden: `disableHostPrefix`.

SDK	Unterstützt	Hinweise oder weitere Informationen
SDK für JavaScript 2.x	Nein	Die Einstellung wird nicht unterstützt, kann aber im Code auf dem Client konfiguriert werden mit: `hostPrefixEnabled` .
SDK für Kotlin	Nein	
SDK for .NET 4.x	Nein	Die Einstellung wird nicht unterstützt, kann aber im Code auf dem Client konfiguriert werden mit: `DisableHostPrefixInjection` .
SDK for .NET 3.x	Nein	Die Einstellung wird nicht unterstützt, kann aber im Code auf dem Client konfiguriert werden mit: `DisableHostPrefixInjection` .
SDK for PHP 3.x	Nein	Die Einstellung wird nicht unterstützt, kann aber im Code auf dem Client konfiguriert werden mit: `disable_host_prefix_injection` .
SDK for Python (Boto3)	Ja	Kann im Code auf dem Client konfiguriert werden mit:. `inject_host_prefix`
SDK for Ruby 3.x	Nein	Die Einstellung wird nicht unterstützt, kann aber im Code auf dem Client konfiguriert werden mit: `disable_host_prefix_injection` .
SDK für Rust	Nein	
SDK für Swift	Nein	
Tools für PowerShell V5	Nein	Die Einstellung wird nicht unterstützt, kann aber mithilfe von Parametern in bestimmte Cmdlets aufgenommen werden. `-ClientConfig @{DisableHostPrefixInjection = $true}`

SDK	U zt	Hinweise oder weitere Informationen
Tools für V4 PowerShell	Neir	Die Einstellung wird nicht unterstützt, kann aber mithilfe von Parametern in bestimmte Cmdlets aufgenommen werden. `-ClientConfig @{DisableHostPrefixInjection = $true}`

IMDS-Kunde

> **Note**
>
> Hilfe zum Verständnis des Layouts von Einstellungsseiten oder zur Interpretation der nachfolgenden Tabelle Support by AWS SDKs and Tools finden Sie unterDie Einstellungsseiten dieses Handbuchs verstehen.

SDKs implementieren einen IMDSv2-Client (Instance Metadata Service Version 2) mithilfe von sitzungsorientierten Anfragen. Weitere Informationen zu IMDSv2 finden Sie unter Verwenden von IMDSv2 im Amazon EC2 EC2-Benutzerhandbuch. Der IMDS-Client ist über ein Client-Konfigurationsobjekt konfigurierbar, das in der SDK-Codebasis verfügbar ist.

Konfigurieren Sie diese Funktionalität wie folgt:

`retries`- Mitglied des Client-Konfigurationsobjekts

Die Anzahl der zusätzlichen Wiederholungsversuche für jede fehlgeschlagene Anfrage.

Standardwert: 3

Gültige Werte: Zahl größer als 0.

`port`- Mitglied des Client-Konfigurationsobjekts

Der Port für den Endpunkt.

Standardwert: 80

Gültige Werte: Zahl.

`token_ttl`- Mitglied des Client-Konfigurationsobjekts

Die TTL des Tokens.

Standardwert: 21.600 Sekunden (6 Stunden, die maximal zugewiesene Zeit).

Gültige Werte: Zahl.

`endpoint`- Mitglied des Client-Konfigurationsobjekts

Der Endpunkt von IMDS.

Standardwert: Wenn `endpoint_mode` gleich`IPv4`, dann ist der Standardendpunkt. `http://169.254.169.254` Wenn `endpoint_mode` gleich`IPv6`, dann ist der Standardendpunkt. `http://[fd00:ec2::254]`

Gültige Werte: Gültiger URI.

Die folgenden Optionen werden von den meisten SDKs unterstützt. Einzelheiten finden Sie in Ihrer spezifischen SDK-Codebasis.

`endpoint_mode`- Mitglied des Client-Konfigurationsobjekts

Der Endpunktmodus von IMDS.

Standardwert: `IPv4`

Zulässige Werte: `IPv4`, `IPv6`

`http_open_timeout`- Mitglied des Client-Konfigurationsobjekts (Name kann variieren)

Die Anzahl der Sekunden, die darauf gewartet werden soll, dass die Verbindung geöffnet wird.

Standardwert: 1 Sekunde.

Gültige Werte: Zahl größer als 0.

`http_read_timeout`- Mitglied des Client-Konfigurationsobjekts (Name kann variieren)

Die Anzahl der Sekunden, für die ein Datenblock gelesen werden muss.

Standardwert: 1 Sekunde.

Gültige Werte: Zahl größer als 0.

`http_debug_output`- Mitglied des Client-Konfigurationsobjekts (Name kann variieren)

Legt einen Ausgabestream für das Debuggen fest.

Standardwert: Keiner.

Gültige Werte: Ein gültiger I/O Stream, wie STDOUT.

`backoff`- Mitglied des Client-Konfigurationsobjekts (Name kann variieren)

Die Anzahl der Sekunden, die zwischen Wiederholungsversuchen oder einem vom Kunden bereitgestellten Backoff-Funktion zum Aufrufen in den Ruhezustand vergehen. Dadurch wird die standardmäßige exponentielle Backoff-Strategie außer Kraft gesetzt.

Standardwert: Variiert je nach SDK.

Gültige Werte: Variiert je nach SDK. Kann entweder ein numerischer Wert oder ein Aufruf einer benutzerdefinierten Funktion sein.

Support durch AWS SDKs und Tools

Die folgenden SDKs unterstützen die in diesem Thema beschriebenen Funktionen und Einstellungen. Alle teilweisen Ausnahmen werden vermerkt. Alle Einstellungen für JVM-Systemeigenschaften werden AWS SDK für Kotlin nur von AWS SDK für Java und vom unterstützt.

SDK	Unterstützt	Hinweise oder weitere Informationen
AWS CLI v2	Ja	
SDK for C++	Nein	
SDK for Go V2 (1.x)	Ja	
SDK for Go 1.x (V1)	Ja	
SDK for Java 2.x	Ja	
SDK for Java 1.x	Ja	

SDK	Unterstützt	Hinweise oder weitere Informationen
SDK für 3.x JavaScript	Ja	
SDK für 2.x JavaScript	Ja	
SDK für Kotlin	Nein	
SDK for .NET 4.x	Ja	
SDK for .NET 3.x	Ja	
SDK for PHP 3.x	Ja	
SDK for Python (Boto3)	Ja	
SDK for Ruby 3.x	Ja	
SDK für Rust	Ja	
SDK für Swift	Ja	
Tools für PowerShell V5	Ja	
Tools für V4 PowerShell	Ja	

Wiederholungsverhalten

> **Important**
>
> Das auf dieser Seite beschriebene Verhalten muss aktiviert werden, bis es zum Standardverhalten wird. `AWS_NEW_RETRIES_2026=true`In Ihrer Umgebung festgelegt. Ohne diese Einstellung verwendet Ihr SDK ein Wiederholungsverhalten aus der Zeit vor 2026, das sich im Backoff-Timing, in den Kosten für Wiederholungsversuche und in den dienstspezifischen Standardeinstellungen unterscheidet. Einzelheiten finden Sie im Blogbeitrag zur Ankündigung.

Wenn eine Anfrage an aufgrund eines vorübergehenden Fehlers oder einer Drosselung AWS-Service fehlschlägt, kann das SDK die Anfrage automatisch wiederholen. Auf dieser Seite wird beschrieben, wie Wiederholungen konfiguriert werden und wie sie intern funktionieren.

- Konfiguration von Wiederholungsversuchen: Wählen Sie einen Wiederholungsmodus, legen Sie die maximale Anzahl der Versuche fest und machen Sie sich mit der Rangfolge der Konfiguration vertraut.
- Wie funktionieren Wiederholungsversuche: Wiederholungsablauf, Fehlerklassifizierung, Backoff-Formel, Mechanismen für Wiederholungsquoten und dienstspezifisches Verhalten.

Konfiguration von Wiederholungsversuchen

Sie steuern, welche Wiederholungsstrategie das SDK verwendet und wie oft es es wiederholt.

Wählen Sie einen Wiederholungsmodus

Der Wiederholungsmodus bestimmt, wie sich das SDK verhält, wenn eine Anfrage fehlschlägt. Drei Modi sind verfügbar: Standard, Adaptiv und Legacy.

	Standard	Adaptiv	Veraltet
Kontingent erneut versuchen	Ja	Ja	Variiert je nach SDK
Kann die erste Anfrage verzögern	Nein	Ja	Nein
Error-type-specific einen Rückzieher	Ja	Ja	Variiert je nach SDK
Für alle SDKs standardisiert	Ja	Ja	Nein
Empfehlung	Standard für alle Workloads	Single-resource, stark drosselnd, latenztolerant	Nur Abwärtskompatibilität

Standardmodus (Standard)

Der Standardmodus wiederholt fehlgeschlagene Anfragen mithilfe eines exponentiellen Backoffs mit Jitter. Er verwendet kürzere Verzögerungen für vorübergehende Fehler (z. B. Netzwerk-Timeouts) und längere Verzögerungen für Drosselungsfehler (z. B.). `ThrottlingException`

Der Standardmodus umfasst ein Wiederholungskontingent, einen Token-Bucket, der bei jedem erneuten Versuch Tokens abzieht und bei erfolgreichen Anfragen die Tokens wieder auffüllt. Wenn die verfügbaren Token erschöpft sind, gibt das SDK den Fehler zurück, ohne es erneut zu versuchen, sodass Ihre Anwendung schnell fehlschlägt, anstatt Wiederholungen abzuwarten, deren Erfolg unwahrscheinlich ist. Dies trägt auch dazu bei, dass Serviceunterbrechungen schneller behoben werden, indem der Datenverkehr mit Wiederholungsversuchen reduziert wird. Während des normalen Betriebs bleibt das Kontingent voll und hat keine Auswirkung. Das Wiederholungskontingent verzögert oder blockiert niemals die erste Anfrage. Nur Wiederholungen sind betroffen. Details hierzu finden Sie unter Kontingent erneut versuchen (Token-Bucket).

Verwenden Sie den Standardmodus, es sei denn, Sie haben einen bestimmten Grund, einen anderen Modus zu wählen.

Adaptiver Modus

Der adaptive Modus umfasst alle Funktionen des Standardmodus sowie einen clientseitigen Ratenbegrenzer. Der Ratenbegrenzer verfolgt Drosselungsreaktionen und passt die Geschwindigkeit an, mit der das SDK Anfragen sendet. Im Gegensatz zum Standardmodus kann der adaptive Modus die erste Anfrage verzögern oder blockieren, nicht nur Wiederholungsversuche, wenn eine Drosselung erkannt wird.

Der Ratenbegrenzer funktioniert pro SDK-Clientinstanz. Für alle Anfragen von einem Client gilt dasselbe Ratenlimit, unabhängig davon, auf welchen API-Vorgang oder welche Ressource sie abzielen.

Wann sollte der adaptive Modus verwendet werden:

- Ihr Client zielt auf eine einzelne Ressource ab (z. B. eine DynamoDB-Tabelle), und Sie erwarten häufige Drosselungsreaktionen. Dies ist häufig bei automatisierten Workflows, Batch-Prozessoren oder KI-Workloads der Fall, bei denen ein einzelner API-Vorgang mit hohem Volumen aufgerufen wird.
- Sie möchten, dass das SDK automatisch langsamer wird, wenn der Dienst eine Drosselung signalisiert.

Wann Sie den adaptiven Modus nicht verwenden sollten:

- Ihr Client sendet Anfragen an mehrere Ressourcen oder bedient mehrere Mandanten. Die Drosselung einer Ressource führt dazu, dass der Ratenbegrenzer alle Anfragen von diesem Client verlangsamt, einschließlich Anfragen an nicht betroffene Ressourcen.
- Sie benötigen eine vorhersehbare Latenz bei der ersten Anfrage.

Der adaptive Modus wird nicht als allgemeine Standardeinstellung empfohlen.

Legacy-Modus

Der Legacy-Modus ist das Wiederholungsverhalten, das jedes SDK vor der Einführung des Standardmodus verwendet hat. Es beinhaltet kein standardisiertes Wiederholungskontingent. Einige SDKs (wie Java) verfügten im Legacy-Modus über eigene Implementierungen für Wiederholungsquoten, aber das Verhalten ist bei allen SDKs nicht einheitlich. Ohne ein standardisiertes Kontingent versucht ein Client bei Serviceunterbrechungen weiterhin mit voller Geschwindigkeit. Dadurch werden Threads und Verbindungen bei Anfragen blockiert, die wahrscheinlich nicht erfolgreich sind, und gleichzeitig wird die Last erhöht, was die Wiederherstellung des Dienstes verzögern kann.

Der Legacy-Modus ist je nach SDKs unterschiedlich. Die Anzahl der Wiederholungen, der Backoff-Timing, die Anzahl wiederholbarer Fehler und das Einschränkungsverhalten unterscheiden sich von Sprache zu Sprache. Code, der vom Verhalten älterer Versionen bei Wiederholungsversuchen abhängt, kann sich anders verhalten, wenn er zwischen SDKs verschoben wird.

Verfügbar in: Java, Python, Ruby, PHP, C++, CLI

Nicht verfügbar in: .NET, Go, Kotlin, Rust, Swift, JavaScript

Aus Gründen der Abwärtskompatibilität gibt es den Legacy-Modus. Wenn Sie derzeit den Legacy-Modus verwenden, wechseln Sie in den Standardmodus.

Versuchen Sie es erneut mit den Einstellungen

Die folgenden Einstellungen steuern das Verhalten bei Wiederholungsversuchen. Sie können sie über Umgebungsvariablen, die gemeinsam genutzte Konfigurationsdatei (`~/.aws/config`) oder die Client-Konfiguration im Code festlegen.

Einstellung	Was es steuert	Umgebungsvariable	Schlüssel der Konfigurationsdatei	Standard
Wiederholungsmodus	Welche Wiederholungsstrategie soll verwendet werden	`AWS_RETRY_MODE`	`retry_mode`	`standard`
Max versucht	Gesamtzahl der Versuche, einschließlich der ersten Anfrage	`AWS_MAX_ATTEMPTS`	`max_attempts`	3(siehe Hinweise)

Ein Wert für max. Versuche 3 bedeutet, dass das SDK eine erste Anfrage und bis zu zwei Wiederholungen durchführt. Legen Sie die maximale Anzahl der Versuche fest, 1 um Wiederholungen vollständig zu deaktivieren.

> **Note**
>
> Die DynamoDB- und DynamoDB Streams Streams-Clients verwenden standardmäßig die maximale Anzahl von Versuchen. 4 Diese Dienste verwenden eine kürzere Basis-Backoff-Verzögerung (25 ms statt 50 ms), um ihrem Profil mit niedriger Latenz zu entsprechen. Durch den zusätzlichen Versuch ist der maximale Backoff-Wert des letzten Wiederholungsversuchs mit dem anderer Dienste vergleichbar. Sie können dies mit den gleichen Einstellungen aus der vorherigen Tabelle außer Kraft setzen.

Priorität der Konfiguration

Wenn Sie dieselbe Einstellung an mehreren Stellen angeben, löst das SDK den Wert anhand der folgenden Rangfolge auf, vom höchsten zum niedrigsten Wert:

1. Explizite Client-Konfiguration im Code. Ein Wert, der direkt auf dem SDK-Client oder seinem Konfigurationsobjekt festgelegt wurde.
2. Umgebungsvariable. Zum Beispiel `AWS_RETRY_MODE` oder `AWS_MAX_ATTEMPTS`.
3. Gemeinsam genutzte Konfigurationsdatei. Geben Sie die `max_attempts` Taste `retry_mode` oder ein`~/.aws/config`.
4. SDK-Standard. Die integrierte Standardeinstellung für die Einstellung.

Dies entspricht der Priorität der AWS Standard-SDK-Konfiguration. Ein auf einer höheren Ebene festgelegter Wert hat immer Vorrang vor einem Wert auf einer niedrigeren Ebene. Wenn Sie beispielsweise den Wert `AWS_RETRY_MODE=adaptive` als Umgebungsvariable und `retry_mode=standard` in festlegen`~/.aws/config`, verwendet das SDK den adaptiven Modus.

Language-specific Konfiguration

Die auf dieser Seite beschriebenen Cross-SDK-Einstellungen (`retry_mode`und`max_attempts`) funktionieren in allen SDKs. Die API für die Konfiguration von Wiederholungsversuchen im Code variiert jedoch je nach Sprache. Im Entwicklerhandbuch Ihres SDK finden Sie sprachspezifische Konfigurationsoptionen wie benutzerdefinierte Backoff-Strategien, zusätzliche Fehler, die wiederholt werden können, und die Kontingentoptimierung für Wiederholungsversuche.

Wie funktionieren Wiederholungsversuche

In diesem Abschnitt wird beschrieben, wie AWS SDKs mit fehlgeschlagenen Anfragen umgehen: Welche Fehler lösen Wiederholungen aus, wie lange das SDK zwischen den Versuchen wartet und wann es die Wiederholungsversuche beendet.

Was passiert, wenn eine Anfrage fehlschlägt

Wenn Sie einen API-Aufruf über ein AWS SDK tätigen, folgt das SDK dieser Reihenfolge:

1. Nur adaptiver Modus: Das SDK überprüft den clientseitigen Ratenbegrenzer. Wenn eine Drosselung erkannt wurde, kann das SDK die Anfrage verzögern oder blockieren, bevor sie gesendet wird.
2. Das SDK sendet die Anfrage an den AWS-Service Endpunkt.
3. Wenn der Dienst eine erfolgreiche Antwort zurückgibt, gibt das SDK das Ergebnis an Ihren Code zurück.
4. Wenn die Anfrage fehlschlägt, stuft das SDK den Fehler als vorübergehend, drosselnd oder nicht wiederholbar ein. Siehe Welche Fehler werden wiederholt.
5. Wenn der Fehler nicht erneut versucht werden kann, gibt das SDK den Fehler sofort an Ihren Code zurück. Es wird kein erneuter Versuch versucht.
6. Wenn der Fehler erneut versucht werden kann, überprüft das SDK, ob die maximale Anzahl von Versuchen erreicht wurde. Wenn ja, gibt es den Fehler an Ihren Code zurück.
7. Das SDK überprüft dieKontingent erneut versuchen (Token-Bucket). Wenn das Token-Budget aufgebraucht ist, versucht das SDK es nicht erneut und gibt den Fehler an Ihren Code zurück.

Ausnahme: Denn das SDK wendet immer noch eine Backoff-Verzögerung anLong-polling Operationen, bevor der Fehler zurückgegeben wird.

8. Das SDK berechnet eine Backoff-Verzögerung auf der Grundlage des Fehlertyps und der Nummer des Wiederholungsversuchs. Siehe Wie lange wartet das SDK.
9. Das SDK wartet auf die berechnete Verzögerung und sendet die Anfrage dann ab Schritt 2 erneut.

Das SDK wiederholt diese Schleife, bis die Anfrage erfolgreich ist, die maximale Anzahl von Versuchen erreicht ist, das Wiederholungskontingent erschöpft ist oder ein Fehler auftritt, der nicht wiederholt werden kann. Der gesamte Vorgang läuft automatisch ab. Bei Ihrer Anwendung wird entweder eine erfolgreiche Antwort oder ein letzter Fehler angezeigt.

Welche Fehler werden wiederholt

Das SDK unterteilt jede fehlgeschlagene Anfrage in eine von drei Kategorien: vorübergehend, drosselnd oder nicht wiederholbar. Diese Klassifizierung bestimmt, ob das SDK die Anfrage erneut versucht und wie lange es wartet, bis es erneut versucht.

Die Klassifizierung basiert auf dem Fehlercode und dem HTTP-Statuscode in der Serviceantwort. Beispielsweise wird ein HTTP 400 mit dem Fehlercode als vorübergehend `RequestTimeout` eingestuft und erneut versucht. Ein HTTP 400 mit `ValidationException` wird als nicht wiederholbar eingestuft und sofort zurückgegeben.

Fehlerklassifizierung

Vorübergehende Fehler werden mit einer kurzen Basisverzögerung (50 ms) wiederholt:

Fehlercode
`RequestTimeout`
`RequestTimeoutException`
`InternalError`
`IDPCommunicationError`
I/O Fehler (Verbindung zurückgesetzt, DNS-Auflösungsfehler, Socket-Timeout)
(beliebiges HTTP 500, 502, 503 oder 504 ohne erkannten Fehlercode)

Drosselungsfehler werden mit einer längeren Basisverzögerung (1.000 ms) wiederholt:

Fehlercode
`Throttling`
`ThrottlingException`
`ThrottledException`
`RequestThrottledException`
`TooManyRequestsException`
`ProvisionedThroughputExceededException`
`TransactionInProgressException`
`LimitExceededException`
`PriorRequestNotComplete`
`RequestThrottled`
`EC2ThrottledException`
`RequestLimitExceeded`
`SlowDown`
`BandwidthLimitExceeded`

Non-retryable Fehler (wie`AccessDeniedException`,`ValidationException`,`ResourceNotFoundException`) werden sofort an Ihren Code zurückgegeben.

> **Note**
>
> Ein HTTP 5XX mit einem Drosselungsfehlercode wird als Drosselungsfehler und nicht als vorübergehender Fehler eingestuft, obwohl 5XX-Fehler normalerweise vorübergehend sind.

Das SDK sucht zuerst nach dem Fehlercode und greift dann auf den HTTP-Statuscode zurück.

Drosselungsfehler bedeuten, dass der Service Ihre Anfrage aufgrund von Ratenbeschränkungen aktiv abgelehnt hat. Daher wartet das SDK länger, bevor es erneut versucht, dem Service Zeit für die Wiederherstellung der Kapazität zu geben. Die spezifischen Verzögerungen finden Sie unterWie lange wartet das SDK.

Wie lange wartet das SDK

Das SDK verwendet exponentiellen Backoff mit vollem Jitter. Im Durchschnitt wartet jeder Wiederholungsversuch länger als der letzte, wobei die Anfragen von mehreren Clients nach dem Zufallsprinzip verteilt werden.

Basisverzögerungen nach Fehlertyp

Die Basisverzögerung hängt davon ab, ob es sich um einen vorübergehenden Fehler oder um einen drosselnden Fehler handelt:

Fehlertyp	Basisverzögerung	Begründung
Vorübergehend (ohne Drosselung)	50 ms	Vorübergehende Fehler werden in der Regel innerhalb von Millisekunden behoben. Eine kurze Basisverzögerung ermöglicht eine schnelle Wiederherstellung.
Drosselung	1.000 ms	Der Service hat für die Anfrage eine Gebührenbegrenzung festgelegt. Eine längere Basisverzögerung gibt Zeit für die Wiederherstellung der Kapazität.

Backoff-Formel

Das SDK berechnet jede Verzögerung beim erneuten Versuch anhand der folgenden Formel:

```
delay = random(0, 1) × min(20,000 ms, base_delay × 2^retry)
```

Wobei Folgendes gilt:

- `random(0, 1)`gibt einen gleichmäßig verteilten Wert zwischen 0 und 1 zurück
- `base_delay`ist 50 ms für vorübergehende Fehler oder 1.000 ms für Drosselungsfehler
- `retry`beginnt bei 0 für den ersten Wiederholungsversuch (den zweiten Anforderungsversuch insgesamt)

Die maximale Backoff-Obergrenze beträgt 20 Sekunden. Keine einzelne Verzögerung überschreitet 20 Sekunden, unabhängig davon, wie viele Versuche unternommen wurden.

Beispiele, die funktioniert haben

Beispiel 1: Vorübergehender Fehler, max. 3 Versuche

Schritt	Was passiert	Delay
Versuch 1	Erste Anfrage. Der Dienst gibt HTTP 503 zurück.	(Keine)
Versuch 2	Das SDK wartet zufällig (0, 50 ms). Der Wiederholungsversuch schlägt mit 503 fehl.	0—50 ms (durchschnittlich ~25 ms)
Versuch 3	Das SDK wartet zufällig (0, 100 ms). Der Wiederholungsversuch ist erfolgreich.	0—100 ms (durchschnittlich ~50 ms)

Die gesamte zusätzliche Latenz beträgt bei beiden Wiederholungen durchschnittlich etwa 75 ms.

Beispiel 2: Drosselungsfehler, max. 3 Versuche

Schritt	Was passiert	Delay
Versuch 1	Erste Anfrage. Der Service gibt 429 `Throttling` zurück.	(Keine)
Versuch 2	Das SDK wartet zufällig (0, 1.000 ms). Retry gibt 429 zurück.	0—1.000 ms (durchschnittlich ~500 ms)
Versuch 3	Das SDK wartet zufällig (0, 2.000 ms). Der Wiederholungsversuch ist erfolgreich.	0—2.000 ms (durchschnittlich ~1.000 ms)

Die gesamte zusätzliche Latenz beträgt bei beiden Wiederholungen durchschnittlich etwa 1.500 ms.

Beispiel 3: Vorübergehender Fehler, bei dem das Backoff-Limit erreicht wird

Bei einer Basisverzögerung von 50 ms würde die berechnete Verzögerung vor dem Capping wie folgt aussehen:

Versuchen Sie es erneut	Berechnete maximale Verzögerung	Nach 20 s Kappe
1	50 ms	50 ms
2	100 ms	100 ms
5	800 ms	800 ms
9	12.800 ms	12.800 ms
10	25.600 ms	20.000 ms

Die Obergrenze wird bei vorübergehenden Fehlern bei der 10. Wiederholung (11. Versuch) wirksam. Bei Drosselungsfehlern mit einer Basisgeschwindigkeit von 1.000 ms wird die Obergrenze bei der sechsten Wiederholung wirksam.

> **Note**
>
> Bei der Standardeinstellung von maximal 3 Versuchen (1 erste Anfrage + 2 Wiederholungen) wird die Backoff-Obergrenze nie erreicht. In dieser Tabelle wird veranschaulicht, was passiert, wenn Sie den Wert `max_attempts` deutlich über den Standardwert hinaus erhöhen.

Warum Jitter wichtig ist

Der Zufallsmultiplikator wird als Full-Jitter bezeichnet. Andernfalls würden alle Clients, bei denen gleichzeitig ein Fehler aufgetreten ist, es gleichzeitig erneut versuchen, was zu einem Anstieg von Wiederholungsverkehr führen würde (das Problem der „donnernden Herde“). Bei vollständigem Jitter werden die Wiederholungsversuche gleichmäßig über das gesamte Backoff-Fenster verteilt, sodass der Service statt synchronisierter Spitzenwerte ein regelmäßiges Rinnsal von Anfragen erhält.

Nehmen wir zum Beispiel an, 1.000 Clients erhalten alle gleichzeitig einen 503. Bei vollständigem Jitter werden die ersten Versuche gleichmäßig über ein 50-ms-Fenster verteilt, anstatt dass alle 1.000 Wiederholungen genau 50 ms dauern.

Server-directed Timing der Wiederholungsversuche

Einige AWS-Services enthalten einen `x-amz-retry-after` Header in Fehlerantworten. Der Header-Wert ist eine Verzögerung in Millisekunden. Wenn dieser Header vorhanden ist, verwendet das SDK die vom Server angegebene Verzögerung, die auf ein Minimum der berechneten Backoff-Verzögerung und ein Maximum der berechneten Backoff-Verzögerung plus 5.000 ms begrenzt ist. Da der berechnete Backoff selbst auf 20 Sekunden begrenzt ist, beträgt die effektive maximale servergesteuerte Verzögerung 25 Sekunden. Das SDK wendet auf diesen Wert keinen Jitter an, da vom Dienst erwartet wird, dass er jitter wird. Auf diese Weise kann der Dienst genau dann kommunizieren, wenn er erwartet, dass Kapazität verfügbar ist.

Kontingent erneut versuchen (Token-Bucket)

Das SDK verwaltet ein internes Token-Budget, das das Verhältnis von erfolgreichen Anfragen zu Fehlschlägen verfolgt. Wenn Fehler weit verbreitet sind, wird das Budget aufgebraucht und das SDK gibt Fehler direkt zurück. Ihre Anwendung schlägt schnell fehl, anstatt Wiederholungsversuche abzuwarten, deren Erfolg unwahrscheinlich ist. Dadurch wird auch der Datenverkehr mit Wiederholungsversuchen reduziert, sodass Serviceunterbrechungen schneller behoben werden können.

So funktioniert das Wiederholungskontingent

Das Token-Budget ist am Anfang voll. Bei jedem erneuten Versuch werden Tokens abgezogen. Wenn ein Wiederholungsversuch erfolgreich ist, stellt das SDK die bei diesem erneuten Versuch verbrauchten Token wieder her. Wenn eine Anfrage beim ersten Versuch erfolgreich ist (keine erneuten Versuche erforderlich), stellt das SDK 1 Token wieder her. Wenn das Budget Null erreicht, hört das SDK auf, es erneut zu versuchen, und gibt Fehler direkt an Ihren Code zurück.

Parameter	Wert
Kapazität des Budgets	500 Wertmarken
Kosten pro vorübergehendem (ohne Drosselung) Wiederholungsversuch	14 Tokens
Kosten pro Drosselungswiederholung	5 Wertmarken

Parameter	Wert
Tokens wurden bei Erfolg nach einem erneuten Versuch wiederhergestellt	Beim letzten Versuch verbrauchte Menge (14 oder 5)
Tokens wurden bei Erfolg ohne erneuten Versuch wiederhergestellt	1 Token

Die höheren Kosten für vorübergehende Wiederholungen sind auf das unterschiedliche Fehlermuster zurückzuführen. Vorübergehende Fehler wie 500er Fehler und Verbindungsausfälle deuten häufig auf ein dienstweites Problem hin. In diesen Situationen ist es unwahrscheinlich, dass ein erneuter Versuch erfolgreich ist. Dies erhöht die Latenz Ihrer Anrufe, bindet Client-Ressourcen und kann die Wiederherstellung für alle verzögern. Drosselungsfehler signalisieren, dass der Dienst mehr Zeit benötigt, bis die Anfrage erfolgreich ausgeführt werden kann. Das SDK wartet zwischen den Wiederholungen länger, um die Erfolgswahrscheinlichkeit zu erhöhen.

Wann blockiert das Kontingent Wiederholungsversuche

Das Wiederholungskontingent verfolgt Tokens jederzeit, blockiert aber nur Wiederholungsversuche, wenn das Budget aufgebraucht ist. Während des normalen Betriebs sind fast alle Anfragen erfolgreich und das Budget bleibt voll. Die Quote hat keine beobachtbaren Auswirkungen auf Wiederholungsversuche.

Bei einem erfolgreichen erneuten Versuch werden nur die eigenen Token-Kosten (14 oder 5 Token) wiederhergestellt, nicht die Kosten früherer fehlgeschlagener Wiederholungen in derselben Anfrage. Wenn beispielsweise der erste Versuch fehlschlägt und der zweite erfolgreich ist, verliert das Budget netto 14 Token. Das Budget wird am schnellsten aufgebraucht, wenn bei Wiederholungsversuchen alle Versuche erschöpft sind, aber es wird auch allmählich aufgebraucht, wenn Anfragen mehrere Wiederholungen erfordern, bevor sie erfolgreich sind.

Bei der Standardeinstellung von maximal 3 Versuchen beginnt das Kontingent zu sinken, wenn mehr als etwa 22% der Anfragen zu dauerhaften vorübergehenden Ausfällen führen, oder mehr als etwa 32% bei Drosselungsfehlern. Unterhalb dieser Raten wird das Budget durch erfolgreiche Anfragen schneller aufgefüllt, als durch fehlgeschlagene Wiederholungen aufgebraucht wird.

Das Startguthaben des Budgets von 500 Token bietet einen Puffer, der kurze Ausfälle auffangen kann. Ein kurzer Anstieg an Fehlern, selbst ein schwerwiegender, blockiert Wiederholungsversuche nicht, es sei denn, er dauert lange genug an, um den Puffer zu leeren.

Praktische Implikationen

- Niedrige Ausfallraten: Die Quote hat keine Wirkung. Das Budget bleibt auf oder nahe der Kapazitätsgrenze.
- Während einer Serviceunterbrechung: Wenn ein hoher Prozentsatz Ihrer Anfragen über einen längeren Zeitraum fehlschlägt, wird das Kontingent aufgebraucht und Ihr Kunde erhält sofort Fehler zurück, anstatt Wiederholungen abzuwarten. Dadurch wird die Latenz auf der Clientseite reduziert, Threads und Verbindungen freigegeben und der Service kann schneller wiederhergestellt werden.
- Wiederherstellung: Wenn der Dienst wiederhergestellt wird und die Anfragen wieder erfolgreich sind, werden bei erfolgreichen Wiederholungsversuchen die vollen Token-Kosten wiederhergestellt, und bei Erfolgen beim ersten Versuch wird 1 Token wiederhergestellt. Das Budget wird schrittweise aufgefüllt und die Wiederholungsversuche werden automatisch fortgesetzt.
- Umfang: Das Token-Budget ist in der Regel auf eine einzelne SDK-Clientinstanz beschränkt. Der genaue Umfang kann je nach SDK variieren. Er wird nicht von Prozessen oder Hosts gemeinsam genutzt.

Service-specific Verhalten

DynamoDB

DynamoDB-Clients verwenden abgestimmte Standardeinstellungen, die für das Profil mit niedriger Latenz von DynamoDB optimiert sind:

Einstellung	Allgemeine Standardeinstellung	DynamoDB-Standard
Vorübergehende Basisverzögerung (ohne Drosselung)	50 ms	25 ms
Basisverzögerung bei Drosselung	1.000 ms	1.000 ms
Max. Versuche	3	4

Diese Standardeinstellungen gelten sowohl für Amazon DynamoDB als auch für DynamoDB Streams.

Long-polling Operationen

Bei bestimmten AWS Vorgängen werden lange Abfragen verwendet. Sie können eine Verbindung offen halten und darauf warten, dass Arbeit eintrifft. Für diese Operationen gilt eine spezielle Wiederholungsbehandlung:

- `SQS.ReceiveMessage`
- `SFN.GetActivityTask`
- `SWF.PollForActivityTask`
- `SWF.PollForDecisionTask`

Spezielles Verhalten: Wenn das Wiederholungskontingent erschöpft ist und Wiederholungsversuche blockiert werden (Schritt 7 inWas passiert, wenn eine Anfrage fehlschlägt), wendet das SDK immer noch eine Backoff-Verzögerung an, bevor der Fehler an Ihren Code zurückgegeben wird.

Dies ist wichtig, da lange Abfragevorgänge normalerweise in einer engen Schleife aufgerufen werden. Ihr Code ruft auf`ReceiveMessage`, verarbeitet alle Nachrichten und ruft `ReceiveMessage` dann sofort wieder auf. Ohne den erzwungenen Backoff würde ein erschöpftes Token-Budget dazu führen, dass das SDK ohne Verzögerung Fehler zurückgibt. Ihre Abfrageschleife würde dann sofort die nächste Anfrage senden, was die CPU-Auslastung des Clients in die Höhe treiben und zusätzlichen Traffic generieren würde. Durch die erzwungene Backoff-Verzögerung wird dieser Zyklus unterbrochen, sodass die Nutzung der Client-Ressourcen und die Abfragerate bei Ausfällen überschaubar bleiben.

Support durch AWS SDKs und Tools

In der folgenden Tabelle ist die Verfügbarkeit des aktualisierten Wiederholungsverhaltens in den einzelnen SDKs aufgeführt. SDK-specific Einzelheiten, einschließlich der Mindestversion, der Standardwerte vor und nach der Aktualisierung und Codebeispiele, finden Sie unter dem Thema Nachverfolgung. GitHub

SDK	Unterstützt	GitHub Problem mit der Nachverfolgung
SDK for Java 2.x	Ja	Problem mit der Nachverfolgung
SDK for Python (Boto3)	Ja	Problem bei der Nachverfolgung
SDK for .NET 4.x	Ja	Problem mit der Nachverfolgung

SDK	Unterstützt	GitHub Problem mit der Nachverfolgung
Tools für PowerShell V5	Ja	Problem mit der Nachverfolgung
SDK für JavaScript 3.x	Ja	Problem mit der Nachverfolgung
SDK for PHP 3.x	Ja	Problem mit der Nachverfolgung
SDK für Kotlin	Ja	Problem mit der Nachverfolgung
SDK für Rust	Ja	Problem mit der Nachverfolgung
SDK für Swift	Siehe Problem mit der Nachverfolgung	Problem mit der Nachverfolgung
SDK for Ruby 3.x	Siehe Problem mit der Nachverfolgung	Problem mit der Nachverfolgung
SDK for Go V2 (1.x)	Siehe Problem mit der Nachverfolgung	Problem mit der Nachverfolgung
SDK for C++	Siehe Problem mit der Nachverfolgung	Problem mit der Nachverfolgung
AWS CLI v2	Siehe Problem mit der Nachverfolgung	Problem mit der Nachverfolgung

Komprimierung anfordern

> **Note**
>
> Hilfe zum Verständnis des Layouts von Einstellungsseiten oder zur Interpretation der nachfolgenden Tabelle Support by AWS SDKs and Tools finden Sie unterDie Einstellungsseiten dieses Handbuchs verstehen.

AWS SDKs und Tools können Payloads automatisch komprimieren, wenn Anfragen an diesen Support gesendet werden AWS-Services , der den Empfang komprimierter Payloads unterstützt. Durch das Komprimieren der Payload auf dem Client vor dem Senden an einen Service können die Gesamtzahl der Anfragen und die Bandbreite, die zum Senden von Daten an den Service erforderlich ist, reduziert werden. Außerdem können erfolglose Anfragen aufgrund von Einschränkungen der Payload-Größe des Dienstes reduziert werden. Für die Komprimierung wählt das SDK oder Tool einen Kodierungsalgorithmus aus, der sowohl vom Dienst als auch vom SDK unterstützt wird. Die aktuelle Liste möglicher Kodierungen besteht jedoch nur aus gzip, kann aber in future erweitert werden.

Die Komprimierung von Anfragen kann besonders nützlich sein, wenn Ihre Anwendung Amazon verwendet CloudWatch. CloudWatch ist ein Überwachungs- und Beobachtungsdienst, der Überwachungs- und Betriebsdaten in Form von Protokollen, Metriken und Ereignissen sammelt. Ein Beispiel für einen Dienstvorgang, der Komprimierung unterstützt, CloudWatch ist die PutMetricDataAPIMethode.

Konfigurieren Sie diese Funktionalität wie folgt:

`disable_request_compression`- Einstellung für gemeinsam genutzte AWS **`config`** Dateien, **`AWS_DISABLE_REQUEST_COMPRESSION`**- Umgebungsvariable, **`aws.disableRequestCompression`**- JVM-Systemeigenschaft: nur Java/Kotlin

Schaltet ein oder aus, ob das SDK oder das Tool eine Nutzlast vor dem Senden einer Anfrage komprimiert.

Standardwert: `false`

Zulässige Werte:

- **`true`**— Deaktiviert die Anforderungskomprimierung.
- **`false`**— Verwenden Sie nach Möglichkeit die Anforderungskomprimierung.

`request_min_compression_size_bytes`- Einstellung für gemeinsam genutzte AWS **`config`** Dateien, **`AWS_REQUEST_MIN_COMPRESSION_SIZE_BYTES`**- Umgebungsvariable, **`aws.requestMinCompressionSizeBytes`**- JVM-Systemeigenschaft: nur Java/Kotlin

Legt die Mindestgröße in Byte des Anforderungstexts fest, den das SDK oder das Tool komprimieren soll. Kleine Payloads können länger werden, wenn sie komprimiert werden. Daher gibt es eine Untergrenze, bei der es sinnvoll ist, eine Komprimierung durchzuführen. Dieser Wert ist inklusiv, eine Anforderungsgröße, die größer oder gleich dem Wert ist, wird komprimiert.

Standardwert: 10240 Byte

Gültige Werte: Ganzzahlwert zwischen 0 und einschließlich 10485760 Byte.

Support durch AWS SDKs und Tools

Die folgenden SDKs unterstützen die in diesem Thema beschriebenen Funktionen und Einstellungen. Alle teilweisen Ausnahmen werden vermerkt. Alle Einstellungen für JVM-Systemeigenschaften werden AWS SDK für Kotlin nur von AWS SDK für Java und vom unterstützt.

SDK	Unterstützt	Hinweise oder weitere Informationen
AWS CLI v2	Ja	
SDK for C++	Ja	
SDK for Go V2 (1.x)	Ja	
SDK for Go 1.x (V1)	Nein	
SDK for Java 2.x	Ja	
SDK for Java 1.x	Nein	
SDK für 3.x JavaScript	Ja	
SDK für 2.x JavaScript	Nein	
SDK für Kotlin	Ja	

SDK	Unterstützt	Hinweise oder weitere Informationen
SDK for .NET 4.x	Ja	
SDK for .NET 3.x	Ja	
SDK for PHP 3.x	Ja	
SDK for Python (Boto3)	Ja	
SDK for Ruby 3.x	Ja	
SDK für Rust	Ja	
SDK für Swift	Nein	
Tools für PowerShell V5	Ja	
Tools für V4 PowerShell	Ja	

Service-specific Endpunkte

> **Note**
>
> Hilfe zum Verständnis des Layouts von Einstellungsseiten oder zur Interpretation der nachfolgenden Tabelle Support by AWS SDKs and Tools finden Sie unterDie Einstellungsseiten dieses Handbuchs verstehen.

Service-specific Die Endpunktkonfiguration bietet die Möglichkeit, einen Endpunkt Ihrer Wahl für API-Anfragen zu verwenden und diese Auswahl beizubehalten. Diese Einstellungen bieten Flexibilität bei der Unterstützung lokaler Endpunkte, VPC-Endpunkte und lokaler AWS -Entwicklungsumgebungen von Drittanbietern. Verschiedene Endpunkte können für Test- und Produktionsumgebungen verwendet werden. Sie können eine Endpunkt-URL für einzelne AWS-Services angeben.

Konfigurieren Sie diese Funktionalität wie folgt:

`endpoint_url`- Einstellung für gemeinsam genutzte AWS **`config`** Dateien, **`AWS_ENDPOINT_URL`**- Umgebungsvariable, **`aws.endpointUrl`**- JVM-Systemeigenschaft: nur Java/Kotlin

Wenn diese Einstellung direkt in einem Profil oder als Umgebungsvariable angegeben wird, gibt sie den Endpunkt an, der für alle Serviceanfragen verwendet wird. Dieser Endpunkt wird von jedem konfigurierten dienstspezifischen Endpunkt überschrieben.

Sie können diese Einstellung auch in einem `services` Abschnitt einer gemeinsam genutzten AWS `config` Datei verwenden, um einen benutzerdefinierten Endpunkt für einen bestimmten Dienst festzulegen. Eine Liste aller Dienstkennungsschlüssel, die für Unterabschnitte innerhalb dieses `services` Abschnitts verwendet werden können, finden Sie unterIdentifikatoren für dienstspezifische Endpunkte.

Standardwert: `none`

Gültige Werte: Eine URL, die das Schema und den Host für den Endpunkt enthält. Die URL kann optional eine Pfadkomponente enthalten, die ein oder mehrere Pfadsegmente enthält.

`AWS_ENDPOINT_URL_<SERVICE>`- Umgebungsvariable, **`aws.endpointUrl<ServiceName>`**- JVM-Systemeigenschaft: nur Java/Kotlin

`AWS_ENDPOINT_URL_<SERVICE>`, wobei sich der AWS-Service Bezeichner `<SERVICE>` befindet, legt einen benutzerdefinierten Endpunkt für einen bestimmten Dienst fest. Eine Liste aller servicespezifischen Umgebungsvariablen finden Sie unter Identifikatoren für dienstspezifische Endpunkte.

Dieser dienstspezifische Endpunkt hat Vorrang vor allen globalen Endpunkten, die in festgelegt sind. `AWS_ENDPOINT_URL`

Standardwert: `none`

Gültige Werte: Eine URL, die das Schema und den Host für den Endpunkt enthält. Die URL kann optional eine Pfadkomponente enthalten, die ein oder mehrere Pfadsegmente enthält.

`ignore_configured_endpoint_urls`- Einstellung für gemeinsam genutzte AWS **`config`** Dateien, **`AWS_IGNORE_CONFIGURED_ENDPOINT_URLS`**- Umgebungsvariable, **`aws.ignoreConfiguredEndpointUrls`**- JVM-Systemeigenschaft: nur Java/Kotlin

Diese Einstellung wird verwendet, um alle benutzerdefinierten Endpunktkonfigurationen zu ignorieren.

Beachten Sie, dass jeder explizite Endpunkt, der im Code oder auf einem Service-Client selbst festgelegt ist, unabhängig von dieser Einstellung verwendet wird. Wenn Sie beispielsweise den `--endpoint-url` Befehlszeilenparameter in einen AWS CLI Befehl aufnehmen oder eine Endpunkt-URL an einen Client-Konstruktor übergeben, ist dies immer wirksam.

Standardwert: `false`

Zulässige Werte:

- **`true`**— Das SDK oder Tool liest keine benutzerdefinierten Konfigurationsoptionen aus der gemeinsam genutzten `config` Datei oder aus Umgebungsvariablen zum Setzen einer Endpunkt-URL.
- **`false`**— Das SDK oder Tool verwendet alle verfügbaren, vom Benutzer bereitgestellten Endpunkte aus der gemeinsam genutzten `config` Datei oder aus Umgebungsvariablen.

Konfigurieren Sie Endpunkte mithilfe von Umgebungsvariablen

Um Anfragen für alle Dienste an eine benutzerdefinierte Endpunkt-URL weiterzuleiten, legen Sie die `AWS_ENDPOINT_URL` globale Umgebungsvariable fest.

```
export AWS_ENDPOINT_URL=http://localhost:4567
```

Verwenden Sie die `AWS_ENDPOINT_URL_<SERVICE>` Umgebungsvariable AWS-Service , um Anfragen für eine bestimmte URL an einen benutzerdefinierten Endpunkt weiterzuleiten. Amazon DynamoDB hat ein `serviceId` von `DynamoDB`. Für diesen Service lautet die Umgebungsvariable für die Endpunkt-URL `AWS_ENDPOINT_URL_DYNAMODB`. Dieser Endpunkt hat Vorrang vor dem globalen Endpunkt, der `AWS_ENDPOINT_URL` für diesen Dienst eingerichtet wurde.

```
export AWS_ENDPOINT_URL_DYNAMODB=http://localhost:5678
```

Als weiteres Beispiel AWS Elastic Beanstalk hat er ein `serviceId` von. `Elastic Beanstalk` Der AWS-Service Bezeichner basiert auf dem API-Modell, indem alle Leerzeichen `serviceId` durch Unterstriche ersetzt und alle Buchstaben in Großbuchstaben geschrieben werden. Um den Endpunkt für diesen Dienst festzulegen, lautet die entsprechende Umgebungsvariable. `AWS_ENDPOINT_URL_ELASTIC_BEANSTALK` Eine Liste aller servicespezifischen Umgebungsvariablen finden Sie unter Identifikatoren für dienstspezifische Endpunkte.

```
export AWS_ENDPOINT_URL_ELASTIC_BEANSTALK=http://localhost:5567
```

Konfigurieren Sie Endpunkte mithilfe von Shared **`config`** file

In der gemeinsam genutzten `config` Datei `endpoint_url` wird es an verschiedenen Stellen für unterschiedliche Funktionen verwendet.

- `endpoint_url`direkt in a angegeben, `profile` macht diesen Endpunkt zum globalen Endpunkt.
- `endpoint_url`Wenn dieser Endpunkt unter einem Dienstbezeichnerschlüssel innerhalb eines `services` Abschnitts verschachtelt ist, gilt dieser Endpunkt nur für Anfragen, die an diesen Dienst gestellt werden. Details zur Definition eines `services`-Abschnitts in Ihrer freigegebenen `config`-Datei finden Sie unter Format der Konfigurationsdatei.

Das folgende Beispiel verwendet eine `services` Definition, um eine dienstspezifische Endpunkt-URL für Amazon S3 und einen benutzerdefinierten globalen Endpunkt für alle anderen Services zu konfigurieren:

```
[profile dev-s3-specific-and-global]
endpoint_url = http://localhost:1234
services = s3-specific

[services s3-specific]
s3 =
  endpoint_url = https://play.min.io:9000
```

Mit einem einzigen Profil können Endpunkte für mehrere Services konfiguriert werden. Dieses Beispiel zeigt, wie die dienstspezifischen Endpunkt-URLs für Amazon S3 und AWS Elastic Beanstalk im selben Profil festgelegt werden. AWS Elastic Beanstalk hat einen `serviceId` von. `Elastic Beanstalk` Der AWS-Service Bezeichner basiert auf dem API-Modell, `serviceId` indem alle Leerzeichen durch Unterstriche ersetzt und alle Buchstaben klein geschrieben werden. Somit wird der Service-Identifier-Schlüssel `elastic_beanstalk` und die Einstellungen für diesen Dienst beginnen in der Zeile. `elastic_beanstalk =` Eine Liste aller Service-ID-Schlüssel, die im `services`-Abschnitt verwendet werden können, finden Sie unter Identifikatoren für dienstspezifische Endpunkte.

```
[services testing-s3-and-eb]
s3 =
  endpoint_url = http://localhost:4567
elastic_beanstalk =
  endpoint_url = http://localhost:8000
```

```
[profile dev]
services = testing-s3-and-eb
```

Der Abschnitt zur Dienstkonfiguration kann von mehreren Profilen aus verwendet werden. Beispielsweise können zwei Profile dieselbe `services` Definition verwenden und gleichzeitig andere Profileigenschaften ändern:

```
[services testing-s3]
s3 =
  endpoint_url = https://localhost:4567

[profile testing-json]
output = json
services = testing-s3

[profile testing-text]
output = text
services = testing-s3
```

Konfigurieren Sie Endpunkte in Profilen mithilfe von rollenbasierten Anmeldeinformationen

Wenn Ihr Profil über rollenbasierte Anmeldeinformationen verfügt, die über einen `source_profile`-Parameter für die IAM-Funktion „Rolle übernehmen“ konfiguriert wurden, verwendet das SDK nur Servicekonfigurationen für das angegebene Profil. Es verwendet keine Profile mit verketteten Rollen. Verwenden Sie beispielsweise die folgende freigegebene `config`-Datei:

```
[profile A]
credential_source = Ec2InstanceMetadata
endpoint_url = https://profile-a-endpoint.aws/

[profile B]
source_profile = A
role_arn = arn:aws:iam::123456789012:role/roleB
services = profileB

[services profileB]
ec2 =
  endpoint_url = https://profile-b-ec2-endpoint.aws
```

Wenn Sie das Profil B verwenden und in Ihrem Code Amazon EC2 aufrufen, wird der Endpunkt als `https://profile-b-ec2-endpoint.aws` aufgelöst. Wenn Ihr Code eine Anforderung für einen anderen Service stellt, folgt die Endpunktauflösung keiner benutzerdefinierten Logik. Der Endpunkt wird nicht zu dem im Profil A definierten globalen Endpunkt aufgelöst. Damit ein globaler Endpunkt für das Profil B wirksam wird, müssten Sie `endpoint_url` direkt im Profil B festlegen. Weitere Informationen zur `source_profile`-Einstellung finden Sie unter Übernehmen Sie die Rolle Credential Provider.

Vorrang der Einstellungen

Die Einstellungen für diese Funktion können gleichzeitig verwendet werden, pro Dienst hat jedoch nur ein Wert Priorität. Für API-Aufrufe an einen bestimmten Wert wird die folgende Reihenfolge verwendet AWS-Service, um einen Wert auszuwählen:

1. Jede explizite Einstellung, die im Code oder auf einem Service-Client selbst festgelegt ist, hat Vorrang vor allen anderen Einstellungen.
   - Für die ist dies der Wert AWS CLI, der vom `--endpoint-url` Befehlszeilenparameter bereitgestellt wird. Bei einem SDK können explizite Zuweisungen die Form eines Parameters annehmen, den Sie festlegen, wenn Sie einen AWS-Service Client oder ein Konfigurationsobjekt instanziieren.
2. Der Wert, der von einer dienstspezifischen Umgebungsvariablen bereitgestellt wird, wie z. `AWS_ENDPOINT_URL_DYNAMODB`
3. Der von der globalen Endpunkt-Umgebungsvariable `AWS_ENDPOINT_URL` bereitgestellte Wert
4. Der Wert, der von der `endpoint_url` Einstellung bereitgestellt wird, die unter einem Dienstbezeichnerschlüssel in einem `services` Abschnitt der gemeinsam genutzten `config` Datei verschachtelt ist.
5. Der Wert, der durch die `endpoint_url` Einstellung bereitgestellt wird, die direkt in einer `profile` der gemeinsam genutzten `config` Datei angegeben wurde.
6. Jede Standard-Endpunkt-URL für die jeweilige AWS-Service Datei wird zuletzt verwendet.

Support durch AWS SDKs und Tools

Die folgenden SDKs unterstützen die in diesem Thema beschriebenen Funktionen und Einstellungen. Alle teilweisen Ausnahmen werden vermerkt. Alle Einstellungen für JVM-Systemeigenschaften werden AWS SDK für Kotlin nur von AWS SDK für Java und vom unterstützt.

SDK	Unterstützt	Hinweise oder weitere Informationen
AWS CLI v2	Ja	
SDK for C++	Ja	
SDK for Go V2 (1.x)	Ja	
SDK for Go 1.x (V1)	Nein	
SDK for Java 2.x	Ja	
SDK for Java 1.x	Nein	
SDK für 3.x JavaScript	Ja	
SDK für 2.x JavaScript	Nein	
SDK für Kotlin	Ja	
SDK for .NET 4.x	Ja	
SDK for .NET 3.x	Ja	
SDK for PHP 3.x	Ja	
SDK for Python (Boto3)	Ja	
SDK for Ruby 3.x	Ja	
SDK für Rust	Ja	
SDK für Swift	Ja	
Tools für PowerShell V5	Ja	
Tools für V4 PowerShell	Ja	

Identifikatoren für dienstspezifische Endpunkte

Informationen darüber, wie und wo Sie die Identifikatoren in der folgenden Tabelle verwenden können, finden Sie unter. Service-specific Endpunkte

serviceId	Di nu üs fü di ge ge D A **c**	**AWS_ENDPOINT_URL_<SERVICE>** Umgebungsvariable
AccessAnalyzer	a l	AWS_ENDPOINT_URL_ACCESSANALYZER
Account	a	AWS_ENDPOINT_URL_ACCOUNT
ACM	a	AWS_ENDPOINT_URL_ACM
ACM PCA	a	AWS_ENDPOINT_URL_ACM_PCA
Alexa For Business	a _	AWS_ENDPOINT_URL_ALEXA_FOR_BUSINESS
amp	a	AWS_ENDPOINT_URL_AMP
Amplify	a	AWS_ENDPOINT_URL_AMPLIFY
AmplifyBackend	a c	AWS_ENDPOINT_URL_AMPLIFYBACKEND
AmplifyUIBuilder	a b	AWS_ENDPOINT_URL_AMPLIFYUIBUILDER

serviceId	Di nu üs fü di ge ge Da AV **co**	**AWS_ENDPOINT_URL_<SERVICE>** Umgebungsvariable
API Gateway	ap ay	AWS_ENDPOINT_URL_API_GATEWAY
ApiGatewayManagemeentApi	ap yr nt	AWS_ENDPOINT_URL_APIGATEWAYMANAGEMENTAPI
ApiGatewayV2	ap yv	AWS_ENDPOINT_URL_APIGATEWAYV2
AppConfig	ap	AWS_ENDPOINT_URL_APPCONFIG
AppConfigData	ap da	AWS_ENDPOINT_URL_APPCONFIGDATA
AppFabric	ap	AWS_ENDPOINT_URL_APPFABRIC
Appflow	ap	AWS_ENDPOINT_URL_APPFLOW
AppIntegrations	ap at	AWS_ENDPOINT_URL_APPINTEGRATIONS
Application Auto Scaling	ap or ca	AWS_ENDPOINT_URL_APPLICATION_AUTO_SCALING

serviceId	Di nu üs fü di ge ge Da AV **co**	**AWS_ENDPOINT_URL_<SERVICE>** Umgebungsvariable
Application Insights	ap or ts	AWS_ENDPOINT_URL_APPLICATION_INSIGHTS
ApplicationCostPro filer	ap or f:	AWS_ENDPOINT_URL_APPLICATIONCOSTPROFILER
App Mesh	ap	AWS_ENDPOINT_URL_APP_MESH
AppRunner	ap	AWS_ENDPOINT_URL_APPRUNNER
AppStream	ap	AWS_ENDPOINT_URL_APPSTREAM
AppSync	ap	AWS_ENDPOINT_URL_APPSYNC
ARC Zonal Shift	a: _s	AWS_ENDPOINT_URL_ARC_ZONAL_SHIFT
Artifact	a:	AWS_ENDPOINT_URL_ARTIFACT
Athena	at	AWS_ENDPOINT_URL_ATHENA
AuditManager	au ge	AWS_ENDPOINT_URL_AUDITMANAGER

serviceId	Di nu üs fü di ge ge Da AV **ce**	**AWS_ENDPOINT_URL_<SERVICE>** Umgebungsvariable
Auto Scaling	au in	AWS_ENDPOINT_URL_AUTO_SCALING
Auto Scaling Plans	au in	AWS_ENDPOINT_URL_AUTO_SCALING_PLANS
b2bi	b2	AWS_ENDPOINT_URL_B2BI
Backup	ba	AWS_ENDPOINT_URL_BACKUP
Backup Gateway	ba te	AWS_ENDPOINT_URL_BACKUP_GATEWAY
BackupStorage	ba ra	AWS_ENDPOINT_URL_BACKUPSTORAGE
Batch	ba	AWS_ENDPOINT_URL_BATCH
BCM Data Exports	bc ex	AWS_ENDPOINT_URL_BCM_DATA_EXPORTS
Bedrock	be	AWS_ENDPOINT_URL_BEDROCK
Bedrock Agent	be ge	AWS_ENDPOINT_URL_BEDROCK_AGENT

serviceId	Di nu üs fü di ge ge D A **co**	**AWS_ENDPOINT_URL_<SERVICE>** Umgebungsvariable
Bedrock Agent Runtime	be ge ir	AWS_ENDPOINT_URL_BEDROCK_AGENT_RUNTIME
Bedrock Runtime	be ur	AWS_ENDPOINT_URL_BEDROCK_RUNTIME
billingconductor	bi nc	AWS_ENDPOINT_URL_BILLINGCONDUCTOR
Braket	br	AWS_ENDPOINT_URL_BRAKET
Budgets	bu	AWS_ENDPOINT_URL_BUDGETS
Cost Explorer	co or	AWS_ENDPOINT_URL_COST_EXPLORER
chatbot	ch	AWS_ENDPOINT_URL_CHATBOT
Chime	ch	AWS_ENDPOINT_URL_CHIME
Chime SDK Identity	ch _i	AWS_ENDPOINT_URL_CHIME_SDK_IDENTITY
Chime SDK Media Pipelines	ch _m pe	AWS_ENDPOINT_URL_CHIME_SDK_MEDIA_PIPELINES

serviceId	Di nu üs fü di ge ge D A **c**	**AWS_ENDPOINT_URL_<SERVICE>** Umgebungsvariable
Chime SDK Meetings	ch _r	AWS_ENDPOINT_URL_CHIME_SDK_MEETINGS
Chime SDK Messaging	ch _r g	AWS_ENDPOINT_URL_CHIME_SDK_MESSAGING
Chime SDK Voice	ch _v	AWS_ENDPOINT_URL_CHIME_SDK_VOICE
CleanRooms	c s	AWS_ENDPOINT_URL_CLEANROOMS
CleanRoomsML	c sr	AWS_ENDPOINT_URL_CLEANROOMSML
Cloud9	c	AWS_ENDPOINT_URL_CLOUD9
CloudControl	c rc	AWS_ENDPOINT_URL_CLOUDCONTROL
CloudDirectory	c ct	AWS_ENDPOINT_URL_CLOUDDIRECTORY
CloudFormation	c at	AWS_ENDPOINT_URL_CLOUDFORMATION

serviceId	Di nu üs fü di ge ge Da AV **co**	**AWS_ENDPOINT_URL_<SERVICE>** Umgebungsvariable
CloudFront	cl t	AWS_ENDPOINT_URL_CLOUDFRONT
CloudFront KeyValueStore	cl t_ es	AWS_ENDPOINT_URL_CLOUDFRONT_KEYVALUESTORE
CloudHSM	cl	AWS_ENDPOINT_URL_CLOUDHSM
CloudHSM V2	cl v2	AWS_ENDPOINT_URL_CLOUDHSM_V2
CloudSearch	cl ch	AWS_ENDPOINT_URL_CLOUDSEARCH
CloudSearch Domain	cl ch	AWS_ENDPOINT_URL_CLOUDSEARCH_DOMAIN
CloudTrail	cl l	AWS_ENDPOINT_URL_CLOUDTRAIL
CloudTrail Data	cl l_	AWS_ENDPOINT_URL_CLOUDTRAIL_DATA
CloudWatch	cl h	AWS_ENDPOINT_URL_CLOUDWATCH

serviceId	Di nu üs fü di ge ge Da AV **co**	**AWS_ENDPOINT_URL_<SERVICE>** Umgebungsvariable
codeartifact	co ac	AWS_ENDPOINT_URL_CODEARTIFACT
CodeBuild	co	AWS_ENDPOINT_URL_CODEBUILD
CodeCatalyst	co ys	AWS_ENDPOINT_URL_CODECATALYST
CodeCommit	co t	AWS_ENDPOINT_URL_CODECOMMIT
CodeDeploy	co y	AWS_ENDPOINT_URL_CODEDEPLOY
CodeGuru Reviewer	co re	AWS_ENDPOINT_URL_CODEGURU_REVIEWER
CodeGuru Security	co se	AWS_ENDPOINT_URL_CODEGURU_SECURITY
CodeGuruProfiler	co ro	AWS_ENDPOINT_URL_CODEGURUPROFILER
CodePipeline	co in	AWS_ENDPOINT_URL_CODEPIPELINE
CodeStar	co	AWS_ENDPOINT_URL_CODESTAR

serviceId	Di nu üs fü di ge ge Da AV **co**	**AWS_ENDPOINT_URL_<SERVICE>** Umgebungsvariable
CodeStar connections	co co ns	AWS_ENDPOINT_URL_CODESTAR_CONNECTIONS
codestar notifications	co no io	AWS_ENDPOINT_URL_CODESTAR_NOTIFICATIONS
Cognito Identity	co de	AWS_ENDPOINT_URL_COGNITO_IDENTITY
Cognito Identity Provider	co de ro	AWS_ENDPOINT_URL_COGNITO_IDENTITY_PROVIDER
Cognito Sync	co yn	AWS_ENDPOINT_URL_COGNITO_SYNC
Comprehend	co d	AWS_ENDPOINT_URL_COMPREHEND
ComprehendMedical	co dr	AWS_ENDPOINT_URL_COMPREHENDMEDICAL
Compute Optimizer	co pt	AWS_ENDPOINT_URL_COMPUTE_OPTIMIZER

serviceId	Di nu üs fü di ge ge Da AV **co**	**AWS_ENDPOINT_URL_<SERVICE>** Umgebungsvariable
Config Service	co ry	AWS_ENDPOINT_URL_CONFIG_SERVICE
Connect	co	AWS_ENDPOINT_URL_CONNECT
Connect Contact Lens	co or ns	AWS_ENDPOINT_URL_CONNECT_CONTACT_LENS
ConnectCampaigns	co mp	AWS_ENDPOINT_URL_CONNECTCAMPAIGNS
ConnectCases	co se	AWS_ENDPOINT_URL_CONNECTCASES
ConnectParticipant	co rt	AWS_ENDPOINT_URL_CONNECTPARTICIPANT
ControlTower	co we	AWS_ENDPOINT_URL_CONTROLTOWER
Cost Optimization Hub	co m: hu	AWS_ENDPOINT_URL_COST_OPTIMIZATION_HUB

serviceId	Di nu üs fü di ge ge Da AV **co**	**AWS_ENDPOINT_URL_<SERVICE>** Umgebungsvariable
Cost and Usage Report Service	co us or ce	AWS_ENDPOINT_URL_COST_AND_USAGE_REPORT_SERVICE
Customer Profiles	cu pr	AWS_ENDPOINT_URL_CUSTOMER_PROFILES
DataBrew	da	AWS_ENDPOINT_URL_DATABREW
DataExchange	da ng	AWS_ENDPOINT_URL_DATAEXCHANGE
Data Pipeline	da li	AWS_ENDPOINT_URL_DATA_PIPELINE
DataSync	da	AWS_ENDPOINT_URL_DATASYNC
DataZone	da	AWS_ENDPOINT_URL_DATAZONE
DAX	da	AWS_ENDPOINT_URL_DAX
Detective	de	AWS_ENDPOINT_URL_DETECTIVE
Device Farm	de rn	AWS_ENDPOINT_URL_DEVICE_FARM

serviceId	Di nu üs fü di ge ge D A **ce**	**AWS_ENDPOINT_URL_<SERVICE>** Umgebungsvariable
DevOps Guru	de ru	AWS_ENDPOINT_URL_DEVOPS_GURU
Direct Connect	di nr	AWS_ENDPOINT_URL_DIRECT_CONNECT
Application Discovery Service	ap or ex ce	AWS_ENDPOINT_URL_APPLICATION_DISCOVERY_SERVICE
DLM	dl	AWS_ENDPOINT_URL_DLM
Database Migration Service	da mi _s	AWS_ENDPOINT_URL_DATABASE_MIGRATION_SERVICE
DocDB	do	AWS_ENDPOINT_URL_DOCDB
DocDB Elastic	do st	AWS_ENDPOINT_URL_DOCDB_ELASTIC
drs	di	AWS_ENDPOINT_URL_DRS
Directory Service	di _s	AWS_ENDPOINT_URL_DIRECTORY_SERVICE
DynamoDB	dy	AWS_ENDPOINT_URL_DYNAMODB

serviceId	Di nu üs fü di ge ge D A **co**	**AWS_ENDPOINT_URL_<SERVICE>** Umgebungsvariable
DynamoDB Streams	dy st	AWS_ENDPOINT_URL_DYNAMODB_STREAMS
EBS	eb	AWS_ENDPOINT_URL_EBS
EC2	ec	AWS_ENDPOINT_URL_EC2
EC2 Instance Connect	ec no ct	AWS_ENDPOINT_URL_EC2_INSTANCE_CONNECT
ECR	ec	AWS_ENDPOINT_URL_ECR
ECR PUBLIC	ec c	AWS_ENDPOINT_URL_ECR_PUBLIC
ECS	ec	AWS_ENDPOINT_URL_ECS
EFS	ef	AWS_ENDPOINT_URL_EFS
EKS	el	AWS_ENDPOINT_URL_EKS
EKS Auth	el	AWS_ENDPOINT_URL_EKS_AUTH
Elastic Inference	el nf	AWS_ENDPOINT_URL_ELASTIC_INFERENCE

serviceId	Di nu üs fü di ge ge Da AV **co**	**AWS_ENDPOINT_URL_<SERVICE>** Umgebungsvariable
ElastiCache	el he	AWS_ENDPOINT_URL_ELASTICACHE
Elastic Beanstalk	el ea	AWS_ENDPOINT_URL_ELASTIC_BEANSTALK
Elastic Transcoder	el ra	AWS_ENDPOINT_URL_ELASTIC_TRANSCODER
Elastic Load Balancing	el oa c:	AWS_ENDPOINT_URL_ELASTIC_LOAD_BALANCING
Elastic Load Balancing v2	el oa c:	AWS_ENDPOINT_URL_ELASTIC_LOAD_BALANCING_V2
EMR	er	AWS_ENDPOINT_URL_EMR
EMR containers	er in	AWS_ENDPOINT_URL_EMR_CONTAINERS
EMR Serverless	er r:	AWS_ENDPOINT_URL_EMR_SERVERLESS

serviceId	Di nu üs fü di ge ge Da AV **co**	**AWS_ENDPOINT_URL_<SERVICE>** Umgebungsvariable
EntityResolution	er oi	AWS_ENDPOINT_URL_ENTITYRESOLUTION
Elasticsearch Service	es an io	AWS_ENDPOINT_URL_ELASTICSEARCH_SERVICE
EventBridge	ev ge	AWS_ENDPOINT_URL_EVENTBRIDGE
Evidently	ev	AWS_ENDPOINT_URL_EVIDENTLY
finspace	fi	AWS_ENDPOINT_URL_FINSPACE
finspace data	fi da	AWS_ENDPOINT_URL_FINSPACE_DATA
Firehose	fi	AWS_ENDPOINT_URL_FIREHOSE
fis	fi	AWS_ENDPOINT_URL_FIS
FMS	fm	AWS_ENDPOINT_URL_FMS
forecast	fo	AWS_ENDPOINT_URL_FORECAST
forecastquery	fo ue	AWS_ENDPOINT_URL_FORECASTQUERY

serviceId	Di nu üs fü di ge ge Da AV **co**	**AWS_ENDPOINT_URL_<SERVICE>** Umgebungsvariable
FraudDetector	fr ct	AWS_ENDPOINT_URL_FRAUDDETECTOR
FreeTier	fr	AWS_ENDPOINT_URL_FREETIER
FSx	fs	AWS_ENDPOINT_URL_FSX
GameLift	ga	AWS_ENDPOINT_URL_GAMELIFT
Glacier	gl	AWS_ENDPOINT_URL_GLACIER
Global Accelerator	gl ce	AWS_ENDPOINT_URL_GLOBAL_ACCELERATOR
Glue	gl	AWS_ENDPOINT_URL_GLUE
grafana	gr	AWS_ENDPOINT_URL_GRAFANA
Greengrass	gr s	AWS_ENDPOINT_URL_GREENGRASS
GreengrassV2	gr sv	AWS_ENDPOINT_URL_GREENGRASSV2
GroundStation	gr tr	AWS_ENDPOINT_URL_GROUNDSTATION

serviceId	Di nu üs fü di ge ge Da AV **co**	**AWS_ENDPOINT_URL_<SERVICE>** Umgebungsvariable
GuardDuty	gu	AWS_ENDPOINT_URL_GUARDDUTY
Health	he	AWS_ENDPOINT_URL_HEALTH
HealthLake	he e	AWS_ENDPOINT_URL_HEALTHLAKE
Honeycode	ho	AWS_ENDPOINT_URL_HONEYCODE
IAM	ia	AWS_ENDPOINT_URL_IAM
identitystore	id to	AWS_ENDPOINT_URL_IDENTITYSTORE
imagebuilder	im de	AWS_ENDPOINT_URL_IMAGEBUILDER
ImportExport	im ox	AWS_ENDPOINT_URL_IMPORTEXPORT
Inspector	in	AWS_ENDPOINT_URL_INSPECTOR
Inspector Scan	in _s	AWS_ENDPOINT_URL_INSPECTOR_SCAN

serviceId	Di nu üs fü di ge ge Da AV **co**	**AWS_ENDPOINT_URL_<SERVICE>** Umgebungsvariable
Inspector2	in 2	AWS_ENDPOINT_URL_INSPECTOR2
InternetMonitor	in or	AWS_ENDPOINT_URL_INTERNETMONITOR
IoT	io	AWS_ENDPOINT_URL_IOT
IoT Data Plane	io p:	AWS_ENDPOINT_URL_IOT_DATA_PLANE
IoT Jobs Data Plane	io da e	AWS_ENDPOINT_URL_IOT_JOBS_DATA_PLANE
IoT 1Click Devices Service	io k_ _s	AWS_ENDPOINT_URL_IOT_1CLICK_DEVICES_ SERVICE
IoT 1Click Projects	io k_ s	AWS_ENDPOINT_URL_IOT_1CLICK_PROJECTS
IoTAnalytics	io io	AWS_ENDPOINT_URL_IOTANALYTICS

serviceId	Di nu üs fü di ge ge Da AV **co**	**AWS_ENDPOINT_URL_<SERVICE>** Umgebungsvariable
IotDeviceAdvisor	io ad	AWS_ENDPOINT_URL_IOTDEVICEADVISOR
IoT Events	io s	AWS_ENDPOINT_URL_IOT_EVENTS
IoT Events Data	io s_	AWS_ENDPOINT_URL_IOT_EVENTS_DATA
IoTFleetHub	io ub	AWS_ENDPOINT_URL_IOTFLEETHUB
IoTFleetWise	io is	AWS_ENDPOINT_URL_IOTFLEETWISE
IoTSecureTunneling	io tu	AWS_ENDPOINT_URL_IOTSECURETUNNELING
IoTSiteWise	io se	AWS_ENDPOINT_URL_IOTSITEWISE
IoTThingsGraph	io gr	AWS_ENDPOINT_URL_IOTTHINGSGRAPH
IoTTwinMaker	io ke	AWS_ENDPOINT_URL_IOTTWINMAKER

serviceId	Di nu üs fü di ge ge Da AV **co**	**AWS_ENDPOINT_URL_<SERVICE>** Umgebungsvariable
IoT Wireless	ic es	AWS_ENDPOINT_URL_IOT_WIRELESS
ivs	iv	AWS_ENDPOINT_URL_IVS
IVS RealTime	iv ir	AWS_ENDPOINT_URL_IVS_REALTIME
ivschat	iv	AWS_ENDPOINT_URL_IVSCHAT
Kafka	ka	AWS_ENDPOINT_URL_KAFKA
KafkaConnect	ka ec	AWS_ENDPOINT_URL_KAFKACONNECT
kendra	ke	AWS_ENDPOINT_URL_KENDRA
Kendra Ranking	ke nl	AWS_ENDPOINT_URL_KENDRA_RANKING
Keyspaces	ke	AWS_ENDPOINT_URL_KEYSPACES
Kinesis	ki	AWS_ENDPOINT_URL_KINESIS

serviceId	Di nu üs fü di ge ge Da AV **co**	**AWS_ENDPOINT_URL_<SERVICE>** Umgebungsvariable
Kinesis Video Archived Media	k: ic iv a	AWS_ENDPOINT_URL_KINESIS_VIDEO_ARCHI VED_MEDIA
Kinesis Video Media	k: ic a	AWS_ENDPOINT_URL_KINESIS_VIDEO_MEDIA
Kinesis Video Signaling	k: ic a:	AWS_ENDPOINT_URL_KINESIS_VIDEO_SIGNALING
Kinesis Video WebRTC Storage	k: ic tc e	AWS_ENDPOINT_URL_KINESIS_VIDEO_WEBRT C_STORAGE
Kinesis Analytics	k: na	AWS_ENDPOINT_URL_KINESIS_ANALYTICS
Kinesis Analytics V2	k: na v2	AWS_ENDPOINT_URL_KINESIS_ANALYTICS_V2
Kinesis Video	k: ic	AWS_ENDPOINT_URL_KINESIS_VIDEO

serviceId	Di nu üs fü di ge ge Da AV **co**	**AWS_ENDPOINT_URL_<SERVICE>** Umgebungsvariable
KMS	kr	AWS_ENDPOINT_URL_KMS
LakeFormation	la t:	AWS_ENDPOINT_URL_LAKEFORMATION
Lambda	la	AWS_ENDPOINT_URL_LAMBDA
Launch Wizard	la za	AWS_ENDPOINT_URL_LAUNCH_WIZARD
Lex Model Building Service	le _l _s	AWS_ENDPOINT_URL_LEX_MODEL_BUILDING_SERVICE
Lex Runtime Service	le me e	AWS_ENDPOINT_URL_LEX_RUNTIME_SERVICE
Lex Models V2	le s_	AWS_ENDPOINT_URL_LEX_MODELS_V2
Lex Runtime V2	le me	AWS_ENDPOINT_URL_LEX_RUNTIME_V2
License Manager	li ar	AWS_ENDPOINT_URL_LICENSE_MANAGER

serviceId	Di nu üs fü di ge ge Da AV **co**	**AWS_ENDPOINT_URL_<SERVICE>** Umgebungsvariable
License Manager Linux Subscriptions	l: ar nu r:	AWS_ENDPOINT_URL_LICENSE_MANAGER_LIN UX_SUBSCRIPTIONS
License Manager User Subscriptions	l: ar e: ip	AWS_ENDPOINT_URL_LICENSE_MANAGER_USE R_SUBSCRIPTIONS
Lightsail	l:	AWS_ENDPOINT_URL_LIGHTSAIL
Location	lo	AWS_ENDPOINT_URL_LOCATION
CloudWatch Logs	c: h_	AWS_ENDPOINT_URL_CLOUDWATCH_LOGS
LookoutEquipment	lo u:	AWS_ENDPOINT_URL_LOOKOUTEQUIPMENT
LookoutMetrics	lo t:	AWS_ENDPOINT_URL_LOOKOUTMETRICS
LookoutVision	lo s:	AWS_ENDPOINT_URL_LOOKOUTVISION

serviceId	Di nu üs fü di ge ge Da AV **ce**	**AWS_ENDPOINT_URL_<SERVICE>** Umgebungsvariable
m2	m2	AWS_ENDPOINT_URL_M2
Machine Learning	ma ea	AWS_ENDPOINT_URL_MACHINE_LEARNING
Macie2	ma	AWS_ENDPOINT_URL_MACIE2
ManagedBlockchain	ma oc	AWS_ENDPOINT_URL_MANAGEDBLOCKCHAIN
ManagedBlockchain Query	ma oc qu	AWS_ENDPOINT_URL_MANAGEDBLOCKCHAIN_QUERY
Marketplace Agreement	ma ce er	AWS_ENDPOINT_URL_MARKETPLACE_AGREEMENT
Marketplace Catalog	ma ce g	AWS_ENDPOINT_URL_MARKETPLACE_CATALOG
Marketplace Deployment	ma ce me	AWS_ENDPOINT_URL_MARKETPLACE_DEPLOYMENT

serviceId	Di nu üs fü di ge ge Da AV **ce**	**AWS_ENDPOINT_URL_<SERVICE>** Umgebungsvariable
Marketplace Entitlement Service	ma ce er v:	AWS_ENDPOINT_URL_MARKETPLACE_ENTITLEMENT_SERVICE
Marketplace Commerce Analytics	ma ce ce io	AWS_ENDPOINT_URL_MARKETPLACE_COMMERCE_ANALYTICS
MediaConnect	me ec	AWS_ENDPOINT_URL_MEDIACONNECT
MediaConvert	me e:	AWS_ENDPOINT_URL_MEDIACONVERT
MediaLive	me	AWS_ENDPOINT_URL_MEDIALIVE
MediaPackage	me ag	AWS_ENDPOINT_URL_MEDIAPACKAGE
MediaPackage Vod	me ag	AWS_ENDPOINT_URL_MEDIAPACKAGE_VOD
MediaPackageV2	me ag	AWS_ENDPOINT_URL_MEDIAPACKAGEV2

serviceId	Di nu üs fü di ge ge Da AV **co**	**AWS_ENDPOINT_URL_<SERVICE>** Umgebungsvariable
MediaStore	me e	AWS_ENDPOINT_URL_MEDIASTORE
MediaStore Data	me e_	AWS_ENDPOINT_URL_MEDIASTORE_DATA
MediaTailor	me o:	AWS_ENDPOINT_URL_MEDIATAILOR
Medical Imaging	me ma	AWS_ENDPOINT_URL_MEDICAL_IMAGING
MemoryDB	me	AWS_ENDPOINT_URL_MEMORYDB
Marketplace Metering	ma ce ng	AWS_ENDPOINT_URL_MARKETPLACE_METERING
Migration Hub	m: _I	AWS_ENDPOINT_URL_MIGRATION_HUB
mgn	mg	AWS_ENDPOINT_URL_MGN
Migration Hub Refactor Spaces	m: _I ct es	AWS_ENDPOINT_URL_MIGRATION_HUB_REFAC TOR_SPACES

serviceId	Di nu üs fü di ge ge Da AV **co**	**AWS_ENDPOINT_URL_<SERVICE>** Umgebungsvariable
MigrationHub Config	mi hu g	AWS_ENDPOINT_URL_MIGRATIONHUB_CONFIG
MigrationHubOrches trator	mi hu ti	AWS_ENDPOINT_URL_MIGRATIONHUBORCHESTRATOR
MigrationHubStrategy	mi hu gy	AWS_ENDPOINT_URL_MIGRATIONHUBSTRATEGY
Mobile	mo	AWS_ENDPOINT_URL_MOBILE
mq	mo	AWS_ENDPOINT_URL_MQ
MTurk	mt	AWS_ENDPOINT_URL_MTURK
MWAA	mw	AWS_ENDPOINT_URL_MWAA
Neptune	ne	AWS_ENDPOINT_URL_NEPTUNE
Neptune Graph	ne ra	AWS_ENDPOINT_URL_NEPTUNE_GRAPH
neptunedata	ne ta	AWS_ENDPOINT_URL_NEPTUNEDATA

serviceId	Di nu üs fü di ge ge Da AV **co**	**AWS_ENDPOINT_URL_<SERVICE>** Umgebungsvariable
Network Firewall	ne i:	AWS_ENDPOINT_URL_NETWORK_FIREWALL
NetworkManager	ne na	AWS_ENDPOINT_URL_NETWORKMANAGER
NetworkMonitor	ne n:	AWS_ENDPOINT_URL_NETWORKMONITOR
nimble	n:	AWS_ENDPOINT_URL_NIMBLE
OAM	oa	AWS_ENDPOINT_URL_OAM
Omics	or	AWS_ENDPOINT_URL_OMICS
OpenSearch	op h	AWS_ENDPOINT_URL_OPENSEARCH
OpenSearchServerless	op hs ss	AWS_ENDPOINT_URL_OPENSEARCHSERVERLESS
OpsWorks	op	AWS_ENDPOINT_URL_OPSWORKS
OpsWorksCM	op m	AWS_ENDPOINT_URL_OPSWORKSCM

serviceId	Di nu üs fü di ge ge Da AV **co**	**AWS_ENDPOINT_URL_<SERVICE>** Umgebungsvariable
Organizations	or ic	AWS_ENDPOINT_URL_ORGANIZATIONS
OSIS	os	AWS_ENDPOINT_URL_OSIS
Outposts	ou	AWS_ENDPOINT_URL_OUTPOSTS
p8data	p8	AWS_ENDPOINT_URL_P8DATA
p8data	p8	AWS_ENDPOINT_URL_P8DATA
Panorama	pa	AWS_ENDPOINT_URL_PANORAMA
Payment Cryptography	pa ry hy	AWS_ENDPOINT_URL_PAYMENT_CRYPTOGRAPHY
Payment Cryptography Data	pa ry hy	AWS_ENDPOINT_URL_PAYMENT_CRYPTOGRAPHY_DATA
Pca Connector Ad	pc ct	AWS_ENDPOINT_URL_PCA_CONNECTOR_AD
Personalize	pe ze	AWS_ENDPOINT_URL_PERSONALIZE

serviceId	Di nu üs fü di ge ge D A **co**	**AWS_ENDPOINT_URL_<SERVICE>** Umgebungsvariable
Personalize Events	pe ze	AWS_ENDPOINT_URL_PERSONALIZE_EVENTS
Personalize Runtime	pe ze e	AWS_ENDPOINT_URL_PERSONALIZE_RUNTIME
PI	p:	AWS_ENDPOINT_URL_PI
Pinpoint	p:	AWS_ENDPOINT_URL_PINPOINT
Pinpoint Email	p: er	AWS_ENDPOINT_URL_PINPOINT_EMAIL
Pinpoint SMS Voice	p: sr	AWS_ENDPOINT_URL_PINPOINT_SMS_VOICE
Pinpoint SMS Voice V2	p: sr _\	AWS_ENDPOINT_URL_PINPOINT_SMS_VOICE_V2
Pipes	p:	AWS_ENDPOINT_URL_PIPES
Polly	pc	AWS_ENDPOINT_URL_POLLY
Pricing	p:	AWS_ENDPOINT_URL_PRICING

serviceId	Di nu üs fü di ge ge Da AV **co**	**AWS_ENDPOINT_URL_<SERVICE>** Umgebungsvariable
PrivateNetworks	p: tv	AWS_ENDPOINT_URL_PRIVATENETWORKS
Proton	p:	AWS_ENDPOINT_URL_PROTON
QBusiness	qb	AWS_ENDPOINT_URL_QBUSINESS
QConnect	qc	AWS_ENDPOINT_URL_QCONNECT
QLDB	qI	AWS_ENDPOINT_URL_QLDB
QLDB Session	qI ic	AWS_ENDPOINT_URL_QLDB_SESSION
QuickSight	qu t	AWS_ENDPOINT_URL_QUICKSIGHT
RAM	ra	AWS_ENDPOINT_URL_RAM
rbin	rb	AWS_ENDPOINT_URL_RBIN
RDS	rc	AWS_ENDPOINT_URL_RDS
RDS Data	rc	AWS_ENDPOINT_URL_RDS_DATA
Redshift	re	AWS_ENDPOINT_URL_REDSHIFT

serviceId	Di nu üs fü di ge ge Da AV **co**	**AWS_ENDPOINT_URL_<SERVICE>** Umgebungsvariable
Redshift Data	re da	AWS_ENDPOINT_URL_REDSHIFT_DATA
Redshift Serverless	re se s	AWS_ENDPOINT_URL_REDSHIFT_SERVERLESS
Rekognition	re or	AWS_ENDPOINT_URL_REKOGNITION
repostspace	re ce	AWS_ENDPOINT_URL_REPOSTSPACE
resiliencehub	re eh	AWS_ENDPOINT_URL_RESILIENCEHUB
Resource Explorer 2	re ex 2	AWS_ENDPOINT_URL_RESOURCE_EXPLORER_2
Resource Groups	re gr	AWS_ENDPOINT_URL_RESOURCE_GROUPS
Resource Groups Tagging API	re gr gg	AWS_ENDPOINT_URL_RESOURCE_GROUPS_TAG GING_API

serviceId	Di nu üs fü di ge ge Da AV **co**	**AWS_ENDPOINT_URL_<SERVICE>** Umgebungsvariable
RoboMaker	ro	AWS_ENDPOINT_URL_ROBOMAKER
RolesAnywhere	ro he	AWS_ENDPOINT_URL_ROLESANYWHERE
Route 53	ro	AWS_ENDPOINT_URL_ROUTE_53
Route53 Recovery Cluster	ro ec lu	AWS_ENDPOINT_URL_ROUTE53_RECOVERY_CLUSTER
Route53 Recovery Control Config	ro ec or nf	AWS_ENDPOINT_URL_ROUTE53_RECOVERY_CO NTROL_CONFIG
Route53 Recovery Readiness	ro ec ea	AWS_ENDPOINT_URL_ROUTE53_RECOVERY_RE ADINESS
Route 53 Domains	ro do	AWS_ENDPOINT_URL_ROUTE_53_DOMAINS
Route53Resolver	ro so	AWS_ENDPOINT_URL_ROUTE53RESOLVER

serviceId	Di nu üs fü di ge ge Da AV **co**	**AWS_ENDPOINT_URL_<SERVICE>** Umgebungsvariable
RUM	ru	AWS_ENDPOINT_URL_RUM
S3	s3	AWS_ENDPOINT_URL_S3
S3 Control	s3 l	AWS_ENDPOINT_URL_S3_CONTROL
S3Outposts	s3 s	AWS_ENDPOINT_URL_S3OUTPOSTS
SageMaker	sa	AWS_ENDPOINT_URL_SAGEMAKER
SageMaker A2I Runtime	sa _a ir	AWS_ENDPOINT_URL_SAGEMAKER_A2I_RUNTIME
Sagemaker Edge	sa _e	AWS_ENDPOINT_URL_SAGEMAKER_EDGE
SageMaker FeatureSt ore Runtime	sa _f to ir	AWS_ENDPOINT_URL_SAGEMAKER_FEATUREST ORE_RUNTIME

serviceId	Di nu üs fü di ge ge Da AV **co**	**AWS_ENDPOINT_URL_<SERVICE>** Umgebungsvariable
SageMaker Geospatial	sa _g a:	AWS_ENDPOINT_URL_SAGEMAKER_GEOSPATIAL
SageMaker Metrics	sa _r	AWS_ENDPOINT_URL_SAGEMAKER_METRICS
SageMaker Runtime	sa _:	AWS_ENDPOINT_URL_SAGEMAKER_RUNTIME
savingsplans	sa ar	AWS_ENDPOINT_URL_SAVINGSPLANS
Scheduler	sc	AWS_ENDPOINT_URL_SCHEDULER
schemas	sc	AWS_ENDPOINT_URL_SCHEMAS
SimpleDB	s:	AWS_ENDPOINT_URL_SIMPLEDB
Secrets Manager	se ar	AWS_ENDPOINT_URL_SECRETS_MANAGER
SecurityHub	se ub	AWS_ENDPOINT_URL_SECURITYHUB
SecurityLake	se al	AWS_ENDPOINT_URL_SECURITYLAKE

serviceId	Di nu üs fü di ge ge Da AV **co**	**AWS_ENDPOINT_URL_<SERVICE>** Umgebungsvariable
ServerlessApplicationRepository	se sa io to	AWS_ENDPOINT_URL_SERVERLESSAPPLICATIONREPOSITORY
Service Quotas	se uo	AWS_ENDPOINT_URL_SERVICE_QUOTAS
Service Catalog	se at	AWS_ENDPOINT_URL_SERVICE_CATALOG
Service Catalog AppRegistry	se at pi	AWS_ENDPOINT_URL_SERVICE_CATALOG_APPREGISTRY
ServiceDiscovery	se so	AWS_ENDPOINT_URL_SERVICEDISCOVERY
SES	se	AWS_ENDPOINT_URL_SES
SESv2	se	AWS_ENDPOINT_URL_SESV2
Shield	sh	AWS_ENDPOINT_URL_SHIELD
signer	si	AWS_ENDPOINT_URL_SIGNER

serviceId	Di nu üs fü di ge ge Da AV **co**	**AWS_ENDPOINT_URL_<SERVICE>** Umgebungsvariable
SimSpaceWeaver	s: ea	AWS_ENDPOINT_URL_SIMSPACEWEAVER
SMS	sr	AWS_ENDPOINT_URL_SMS
Snow Device Managemen t	sr ce me	AWS_ENDPOINT_URL_SNOW_DEVICE_MANAGEMENT
Snowball	sr	AWS_ENDPOINT_URL_SNOWBALL
SNS	sr	AWS_ENDPOINT_URL_SNS
SQS	sc	AWS_ENDPOINT_URL_SQS
SSM	ss	AWS_ENDPOINT_URL_SSM
SSM Contacts	ss ct	AWS_ENDPOINT_URL_SSM_CONTACTS
SSM Incidents	ss er	AWS_ENDPOINT_URL_SSM_INCIDENTS
Ssm Sap	ss	AWS_ENDPOINT_URL_SSM_SAP
SSO	ss	AWS_ENDPOINT_URL_SSO

serviceId	Di nu üs fü di ge ge Da AV **co**	**AWS_ENDPOINT_URL_<SERVICE>** Umgebungsvariable
SSO Admin	ss	AWS_ENDPOINT_URL_SSO_ADMIN
SSO OIDC	ss	AWS_ENDPOINT_URL_SSO_OIDC
SFN	st	AWS_ENDPOINT_URL_SFN
Storage Gateway	st at	AWS_ENDPOINT_URL_STORAGE_GATEWAY
STS	st	AWS_ENDPOINT_URL_STS
SupplyChain	su in	AWS_ENDPOINT_URL_SUPPLYCHAIN
Support	su	AWS_ENDPOINT_URL_SUPPORT
Support App	su pp	AWS_ENDPOINT_URL_SUPPORT_APP
SWF	sw	AWS_ENDPOINT_URL_SWF
synthetics	sy s	AWS_ENDPOINT_URL_SYNTHETICS
Textract	te	AWS_ENDPOINT_URL_TEXTRACT

serviceId	Di nu üs fü di ge ge Da AV **co**	**AWS_ENDPOINT_URL_<SERVICE>** Umgebungsvariable
Timestream InfluxDB	t: m_ b	AWS_ENDPOINT_URL_TIMESTREAM_INFLUXDB
Timestream Query	t: m_	AWS_ENDPOINT_URL_TIMESTREAM_QUERY
Timestream Write	t: m_	AWS_ENDPOINT_URL_TIMESTREAM_WRITE
tnb	tr	AWS_ENDPOINT_URL_TNB
Transcribe	t: e	AWS_ENDPOINT_URL_TRANSCRIBE
Transfer	t:	AWS_ENDPOINT_URL_TRANSFER
Translate	t:	AWS_ENDPOINT_URL_TRANSLATE
TrustedAdvisor	t: v:	AWS_ENDPOINT_URL_TRUSTEDADVISOR
VerifiedPermissions	ve e: s	AWS_ENDPOINT_URL_VERIFIEDPERMISSIONS
Voice ID	vo	AWS_ENDPOINT_URL_VOICE_ID

serviceId	Di nu üs fü di ge ge Da AV **co**	**AWS_ENDPOINT_URL_<SERVICE>** Umgebungsvariable
VPC Lattice	vp ce	AWS_ENDPOINT_URL_VPC_LATTICE
WAF	wa	AWS_ENDPOINT_URL_WAF
WAF Regional	wa na	AWS_ENDPOINT_URL_WAF_REGIONAL
WAFV2	wa	AWS_ENDPOINT_URL_WAFV2
WellArchitected	we te	AWS_ENDPOINT_URL_WELLARCHITECTED
Wisdom	wi	AWS_ENDPOINT_URL_WISDOM
WorkDocs	wo	AWS_ENDPOINT_URL_WORKDOCS
WorkLink	wo	AWS_ENDPOINT_URL_WORKLINK
WorkMail	wo	AWS_ENDPOINT_URL_WORKMAIL
WorkMailMessageFlow	wo es w	AWS_ENDPOINT_URL_WORKMAILMESSAGEFLOW
WorkSpaces	wo s	AWS_ENDPOINT_URL_WORKSPACES

serviceId	Di nu üs fü di ge ge D A **co**	**AWS_ENDPOINT_URL_<SERVICE>** Umgebungsvariable
WorkSpaces Thin Client	wo s_ ie	AWS_ENDPOINT_URL_WORKSPACES_THIN_CLIENT
WorkSpaces Web	wo s_	AWS_ENDPOINT_URL_WORKSPACES_WEB
XRay	x:	AWS_ENDPOINT_URL_XRAY

Standardeinstellungen für intelligente Konfigurationen

> **Note**
>
> Hilfe zum Verständnis des Layouts von Einstellungsseiten oder zur Interpretation der nachfolgenden Tabelle Support by AWS SDKs and Tools finden Sie unterDie Einstellungsseiten dieses Handbuchs verstehen.

Mit der Funktion für intelligente Konfigurationsstandardwerte können AWS SDKs vordefinierte, optimierte Standardwerte für andere Konfigurationseinstellungen bereitstellen.

Konfigurieren Sie diese Funktionalität wie folgt:

`defaults_mode`- Einstellung für gemeinsam genutzte AWS **`config`** Dateien, **`AWS_DEFAULTS_MODE`**- Umgebungsvariable, **`aws.defaultsMode`**- JVM-Systemeigenschaft: nur Java/Kotlin

Mit dieser Einstellung können Sie einen Modus wählen, der zu Ihrer Anwendungsarchitektur passt und dann optimierte Standardwerte für Ihre Anwendung bereitstellt. Wenn für eine AWS SDK-Einstellung ein Wert explizit festgelegt ist, hat dieser Wert immer Vorrang. Wenn für eine AWS SDK-Einstellung kein explizit festgelegter Wert festgelegt wurde und sie nicht dem `defaults_mode` Wert einer Legacy-Einstellung entspricht, kann diese Funktion unterschiedliche Standardwerte für verschiedene Einstellungen bereitstellen, die für Ihre Anwendung optimiert sind. Zu den Einstellungen können Folgendes gehören: HTTP-Kommunikationseinstellungen, Wiederholungsverhalten, Einstellungen für den regionalen Endpunkt des Dienstes und möglicherweise jede SDK-related Konfiguration. Kunden, die diese Funktion verwenden, können neue Standardkonfigurationen erhalten, die auf allgemeine Nutzungsszenarien zugeschnitten sind. Wenn Ihre nicht identisch `defaults_mode` ist, empfehlen wir`legacy`, Tests Ihrer Anwendung durchzuführen, wenn Sie das SDK aktualisieren, da sich die angegebenen Standardwerte ändern können, wenn sich die bewährten Methoden weiterentwickeln.

Standardwert: `legacy`

Hinweis: Neue Hauptversionen von SDKs werden standardmäßig verwendet. `standard`

Zulässige Werte:

- `legacy`— Stellt Standardeinstellungen bereit, die je nach SDK variieren und vor der Einrichtung von `defaults_mode` existierten.
- `standard`— Stellt die neuesten empfohlenen Standardwerte bereit, deren Ausführung in den meisten Szenarien sicher sein sollte.
- `in-region`— Baut auf dem Standardmodus auf und beinhaltet eine Optimierung, die auf Anwendungen zugeschnitten ist, die AWS-Services aus demselben Modus heraus aufrufen AWS-Region.
- `cross-region`— Baut auf dem Standardmodus auf und beinhaltet eine Optimierung, die auf Anwendungen zugeschnitten ist, die AWS-Services in einer anderen Region anrufen.
- `mobile`— Baut auf dem Standardmodus auf und beinhaltet eine auf mobile Anwendungen zugeschnittene Optimierung.
- `auto` – Baut auf dem Standardmodus auf und beinhaltet experimentelle Features. Das SDK versucht, die Laufzeitumgebung zu ermitteln, um die entsprechenden Einstellungen automatisch zu ermitteln. Die auto Erkennung basiert auf Heuristik und bietet keine

hundertprozentige Genauigkeit. Wenn die Laufzeitumgebung nicht bestimmt werden kann, `standard` wird der Modus verwendet. Die auto Erkennung fragt möglicherweise Instanzmetadaten ab, was zu Latenz führen kann. Wenn die Startlatenz für Ihre Anwendung entscheidend ist, empfehlen wir, stattdessen einen expliziten `defaults_mode` zu wählen.

Beispiel für die Einstellung dieses Werts in der `config` Datei:

```
[default]
defaults_mode = standard
```

Die folgenden Parameter können basierend auf der Auswahl von optimiert werden`defaults_mode`:

- `retryMode`— Gibt an, wie das SDK versucht, es erneut zu versuchen. Siehe Wiederholungsverhalten.
- `stsRegionalEndpoints`— Gibt an, wie das SDK den AWS-Service Endpunkt bestimmt, über den es mit AWS -Security-Token-Service (AWS STS) kommuniziert. Siehe AWS STS Regionale -Endpunkte.
- `s3UsEast1RegionalEndpoints`— Gibt an, wie das SDK den AWS Service-Endpunkt bestimmt, den es für die Kommunikation mit Amazon S3 für die `us-east-1` Region verwendet.
- `connectTimeoutInMillis`— Nach einem ersten Verbindungsversuch auf einem Socket, die Zeitspanne bis zum Timeout. Wenn der Client den Abschluss des Connect-Handshakes nicht erhält, gibt der Client auf und schlägt den Vorgang fehl.
- `tlsNegotiationTimeoutInMillis`— Die maximale Zeit, die ein TLS-Handshake vom Senden der CLIENT HELLO-Nachricht bis zu dem Zeitpunkt in Anspruch nehmen kann, zu dem der Client und der Server die Chiffren vollständig ausgehandelt und Schlüssel ausgetauscht haben.

Der Standardwert für jede Einstellung ändert sich je nach den Einstellungen, die Sie für Ihre Anwendung `defaults_mode` ausgewählt haben. Diese Werte sind derzeit wie folgt festgelegt (Änderungen vorbehalten):

Parameter	Modus **`standard`**	Modus **`in-region`**	Modus **`cross-region`**	Modus **`mobile`**
`retryMode`	`standard`	`standard`	`standard`	`standard`

Parameter	Modus **standard**	Modus **in-region**	Modus **cross-region**	Modus **mobile**
`stsRegionalEndpoints`	`regional`	`regional`	`regional`	`regional`
`s3UsEast1RegionalEndpoints`	`regional`	`regional`	`regional`	`regional`
`connectTimeoutInMillis`	3100	1100	3100	30000
`tlsNegotiationTimeoutInMillis`	3100	1100	3100	30000

Wenn `defaults_mode` Sie beispielsweise „"“ ausgewählt haben`standard`, wird der `standard` Wert für `retry_mode` (aus den gültigen `retry_mode` Optionen) und der `regional` Wert für `stsRegionalEndpoints` (aus den gültigen `stsRegionalEndpoints` Optionen) zugewiesen.

Support durch AWS SDKs und Tools

Die folgenden SDKs unterstützen die in diesem Thema beschriebenen Funktionen und Einstellungen. Alle teilweisen Ausnahmen werden vermerkt. Alle Einstellungen für JVM-Systemeigenschaften werden AWS SDK für Kotlin nur von AWS SDK für Java und vom unterstützt.

SDK	Unterstützt	Hinweise oder weitere Informationen
AWS CLI v2	Nein	
SDK for C++	Ja	Parameter nicht optimiert :`stsRegionalEndpoin`

SDK	Unterstützt	Hinweise oder weitere Informationen
		`ts` ,`s3UsEast1RegionalEndpoints` ,`tlsNegotiationTimeoutInMillis` .
SDK for Go V2 (1.x)	Ja	Parameter nicht optimiert :`retryMode` ,`stsRegionalEndpoints` ,`s3UsEast1RegionalEndpoints` .
SDK for Go 1.x (V1)	Nein	
SDK for Java 2.x	Ja	Parameter nicht optimiert :`stsRegionalEndpoints` .
SDK for Java 1.x	Nein	
SDK für 3.x JavaScript	Ja	Parameter nicht optimiert :`stsRegionalEndpoints` ,`s3UsEast1RegionalEndpoints` ,`tlsNegotiationTimeoutInMillis` . `connectTimeoutInMillis` heißt`connectionTimeout` .
SDK für JavaScript 2.x	Nein	
SDK für Kotlin	Nein	

SDK	Unterstützt	Hinweise oder weitere Informationen
SDK for .NET 4.x	Ja	Parameter nicht optimiert :`connectTimeoutInMillis` ,`tlsNegotiationTimeoutInMillis` .
SDK for .NET 3.x	Ja	Parameter nicht optimiert :`connectTimeoutInMillis` ,`tlsNegotiationTimeoutInMillis` .
SDK for PHP 3.x	Ja	Parameter nicht optimiert :`tlsNegotiationTimeoutInMillis` .
SDK for Python (Boto3)	Ja	Parameter nicht optimiert:. `tlsNegotiationTimeoutInMillis`
SDK for Ruby 3.x	Ja	
SDK für Rust	Nein	
SDK für Swift	Nein	
Tools für PowerShell V5	Ja	Parameter nicht optimiert :`connectTimeoutInMillis` ,`tlsNegotiationTimeoutInMillis` .

SDK	Unterstützt	Hinweise oder weitere Informationen
Tools für PowerShell V4	Ja	Parameter nicht optimiert :`connectTimeoutInMillis` ,`tlsNegotiationTimeoutInMillis` .

Paginierung

Viele AWS Serviceoperationen können mehr Ergebnisse liefern, als in eine einzige Antwort passen. In diesem Fall enthält die Antwort ein Paginierungstoken, das angibt, dass mehr Ergebnisse verfügbar sind. Um die nächste Ergebnisseite abzurufen, fügen Sie dieses Token in eine nachfolgende Anfrage ein. Dieser Vorgang wird wiederholt, bis kein Token zurückgegeben wird, was bedeutet, dass Sie alle Ergebnisse erhalten haben.

Wie funktioniert die Paginierung

Wenn Sie eine paginierte API-Operation aufrufen, gibt der Dienst eine feste Höchstanzahl von Elementen pro Antwort zurück. Wenn die Gesamtzahl der Elemente dieses Maximum überschreitet, nimmt der Dienst ein Paginierungstoken in die Antwort auf. Der spezifische Name dieses Tokens variiert je nach Dienst. Zu den gebräuchlichen Namen gehören`NextToken`,`Marker`, `ContinuationToken` und. `ExclusiveStartKey` Um die nächste Seite anzufordern, übergeben Sie diesen Tokenwert im entsprechenden Anforderungsparameter Ihres nächsten Aufrufs.

Die meisten paginierten Operationen unterstützen auch einen Parameter, der die maximale Anzahl von Elementen steuert, die pro Seite zurückgegeben werden. Zu den gebräuchlichen Bezeichnungen für diesen Parameter gehören `MaxResults`,`MaxItems`,`MaxKeys`,`MaxRecords`, und`Limit`.

Ein Paginierungstoken ist nur gültig, wenn Sie es in einer früheren Antwort desselben Vorgangs erhalten haben. Wenn die Antwort kein Token enthält, können keine weiteren Ergebnisse abgerufen werden.

Verwenden von Paginatoren

Anstatt Paginierungstoken manuell zu verfolgen, bieten alle AWS SDKs Paginator-Abstraktionen. Ein Paginator kümmert sich automatisch um die Tokenverwaltung, führt aufeinanderfolgende API-Aufrufe

durch und liefert Ergebnisse, bis alle Seiten abgerufen wurden. Die Verwendung eines Paginators vereinfacht Ihren Code und reduziert das Risiko von Fehlern, z. B. wenn Sie vergessen, das Token zu übergeben, oder es falsch verwenden.

Hauptmerkmale von Paginatoren in allen SDKs:

- Paginatoren akzeptieren dieselben Eingabeparameter wie die zugrunde liegende API-Operation.
- Paginatoren erkennen automatisch, wenn mehr Seiten verfügbar sind, und rufen sie ab.
- Paginatoren stellen keine Anfragen mehr, wenn in der Antwort kein Paginierungstoken zurückgegeben wird.
- Einige SDKs bieten zusätzlich Mechanismen zur Begrenzung der Gesamtzahl der abgerufenen Seiten oder Elemente.

Informationen zur Paginator-Syntax und zu den in Ihrer Sprache verfügbaren Verwendungsmustern finden Sie im Entwicklerhandbuch Ihres jeweiligen SDK.

Support durch AWS SDKs und Tools

Die folgenden SDKs unterstützen die in diesem Thema beschriebenen Funktionen und Einstellungen. Alle teilweisen Ausnahmen werden vermerkt. Alle Einstellungen für JVM-Systemeigenschaften werden AWS SDK für Kotlin nur von AWS SDK für Java und vom unterstützt.

SDK	Unterstützt	Hinweise oder weitere Informationen
AWS CLI v2	Ja	Built-in automatische Paginierung für alle Befehle. Zum Deaktivieren verwenden `--no-paginate` .
SDK for C++	Nein	Manuelle Token-Behandlung erforderlich.
SDK for Go V2 (1.x)	Ja	Verwenden Sie `New<OperationName>Paginator` Typen.
SDK for Go 1.x (V1)	Ja	Verwenden Sie `<OperationName>Pages` Methoden.
SDK for Java 2.x	Ja	Verwenden Sie `<operationName>Paginator` Methoden auf dem Client.

SDK	Unterstützt	Hinweise oder weitere Informationen
SDK for Java 1.x	Ja	
SDK für 3.x JavaScript	Ja	Verwenden Sie `paginate<OperationName>` Funktionen.
SDK für JavaScript 2.x	Nein	Manuelle Token-Behandlung erforderlich.
SDK für Kotlin	Ja	Verwenden Sie `<operationName>Paginated` Erweiterungsfunktionen.
SDK for .NET 4.x	Ja	Verwenden Sie `IPaginator` Schnittstellen.
SDK for .NET 3.x	Ja	Verwenden Sie `IPaginator` Schnittstellen.
SDK for PHP 3.x	Ja	`getPaginator` Auf dem Client verwenden.
SDK for Python (Boto3)	Ja	`get_paginator` Auf dem Client verwenden.
SDK for Ruby 3.x	Ja	Wird für Antworten verwendet`.each`, die Paginierung unterstützen.
SDK für Rust	Ja	Verwenden Sie `.into_paginator()` auf Anfrage Builder.
SDK für Swift	Ja	Verwenden Sie `PaginatedSequence` Typen.
Tools für PowerShell V5	Ja	Built-in automatische Paginierung. Zum Deaktivieren verwenden`-NoAutoIteration` .
Tools für PowerShell V4	Ja	

AWS Sicherheitseinstellungen für SDKs und Tools

In diesem Abschnitt werden einige Sicherheitseinstellungen behandelt, die allen AWS SDKs gemeinsam sind. Umfassende Informationen zur Sicherheit einzelner AWS SDKs finden Sie jedoch im entsprechenden AWS SDK-Entwicklerhandbuch im entsprechenden Sicherheitskapitel:

- AWS Kapitel SDK for C++ C++-Sicherheit
- AWS Kapitel SDK for Go Security
- AWS Kapitel SDK for Java Java-Sicherheit
- AWS Kapitel SDK für JavaScript Sicherheit
- AWS Kapitel SDK für Kotlin-Sicherheit
- AWS Kapitel SDK for .NET .NET-Sicherheit
- AWS Kapitel SDK for PHP PHP-Sicherheit
- AWS Kapitel SDK für Python-Sicherheit
- AWS Kapitel SDK for Ruby Security
- AWS Kapitel SDK für Rust Security
- AWS Kapitel SDK für Swift Security

Themen

- Aktivierung von hybridem Post-Quantum-TLS

Aktivierung von hybridem Post-Quantum-TLS

AWS SDKs und Tools verfügen über kryptografische Funktionen und Konfigurationen, die sich je nach Sprache und Laufzeit unterscheiden. Derzeit gibt es drei Möglichkeiten, wie ein AWS SDK oder Tool PQ-TLS-Unterstützung bietet:

Themen

- SDKs mit standardmäßig aktiviertem PQ TLS
- Opt-in PQ TLS-Unterstützung
- SDKs, die auf System OpenSSL basieren

- AWS SDKs und Tools planen nicht, PQ TLS zu unterstützen

SDKs mit standardmäßig aktiviertem PQ TLS

> **Note**
>
> Ab Version 6 AWS verwenden SDK und die zugrunde liegenden CRT-Bibliotheken für macOS und Windows Systembibliotheken für TLS, sodass die PQ-TLS-Funktionen auf diesen Plattformen im Allgemeinen durch die Unterstützung auf Systemebene bestimmt werden. Nov-2025

AWS SDK für Go

Das AWS SDK for Go verwendet Golangs eigene TLS-Implementierung, die in seiner Standardbibliothek bereitgestellt wird. Golang unterstützt und bevorzugt PQ TLS ab Version 1.24, sodass AWS SDK for Go Go-Benutzer PQ TLS aktivieren können, indem sie einfach Golang auf Version 1.24 aktualisieren.

AWS SDK für (Browser) JavaScript

Das AWS SDK für JavaScript (Browser) verwendet den TLS-Stack des Browsers, sodass das SDK PQ-TLS aushandelt, wenn die Browser-Laufzeit dies unterstützt und bevorzugt. Firefox hat die Unterstützung für PQ TLS in Version 132.0 eingeführt. Chrome kündigte die Unterstützung für PQ TLS in Version 131 an. Edge unterstützt Opt-in-PQ TLS in Version 120 für Desktop und Version 140 für Android.

AWS SDK für Node.js

Ab Version Node.js 22.20 (LTS) und Version 24.9.0 verknüpft und bündelt OpenSSL 3.5 Node.js statisch. Das bedeutet, dass PQ TLS für diese und nachfolgende Versionen standardmäßig aktiviert und bevorzugt ist.

AWS SDK für Kotlin

Das Kotlin SDK unterstützt und bevorzugt PQ TLS unter Linux ab Version 1.5.78. Da AWS das SDK für den CRT-based Kotlin-Client auf Systembibliotheken für TLS auf macOS und Windows basiert, hängt die Unterstützung für PQ TLS von diesen zugrunde liegenden Systembibliotheken ab.

AWS SDK für Rust

Das AWS SDK für Rust verteilt unterschiedliche Pakete (im Rust-Ökosystem als „Crates“ bezeichnet) für jeden Service-Client. Diese werden alle in einem konsolidierten GitHub Repository verwaltet, aber jeder Service-Client folgt seiner eigenen Version und seinem eigenen Release-Rhythmus. Das konsolidierte SDK hat die PQ-TLS-Präferenz am 8/29 /25 veröffentlicht, sodass jede einzelne Service Client-Version, die nach diesem Datum veröffentlicht wird, PQ TLS standardmäßig unterstützt und bevorzugt.

Sie können die Mindestversion, die PQ TLS für einen bestimmten Service Client unterstützt, ermitteln, indem Sie zur entsprechenden crates.io-Versions-URL navigieren (zum Beispiel „ AWS-Werbegutschriften's ist hier“) und die erste Version suchen, die nach 29- veröffentlicht wurde. Aug-25 Für jede Service Client-Version, die nach 29 veröffentlicht Aug-25 wird, ist PQ TLS standardmäßig aktiviert und bevorzugt.

Opt-in PQ TLS-Unterstützung

AWS SDK für C++

Standardmäßig verwendet das C++-SDK plattformnative Clients wie libcurl und. WinHttp Libcurl stützt sich im Allgemeinen auf System-OpenSSL für TLS, sodass PQ TLS standardmäßig nur aktiviert ist, wenn System-OpenSSL ≥ v3.5 ist. Sie können diese Standardeinstellung in C++ SDK v1.11.673 oder höher außer Kraft setzen und sich für die Option entscheiden, die PQ TLS standardmäßig unterstützt und aktiviert. AwsCrtHttpClient

Hinweise zum Erstellen für Opt-In PQ TLS Mit diesem Skript können Sie die CRT-Abhängigkeiten des SDK abrufen. Das Erstellen des SDK aus dem Quellcode wird hier und hier beschrieben. Beachten Sie jedoch, dass Sie möglicherweise einige zusätzliche CMake-Flags benötigen:

```
-DUSE_CRT_HTTP_CLIENT=ON \
-DUSE_TLS_V1_2=OFF \
-DUSE_TLS_V1_3=ON \
-DUSE_OPENSSL=OFF \
```

AWS SDK für Java

Ab Version 2 bietet AWS das SDK for Java einen AWS Common Runtime (AWS CRT) -HTTP-Client, der für die Ausführung von PQ-TLS konfiguriert werden kann. Ab Version 2.35.11 AwsCrtHttpClient aktiviert und bevorzugt der standardmäßig PQ TLS, wo immer er verwendet wird.

SDKs, die auf System OpenSSL basieren

Verschiedene AWS SDKs und Tools hängen von der libcrypto/libssl TLS-Bibliothek des Systems ab. Die am häufigsten verwendete Systembibliothek ist OpenSSL. OpenSSL aktivierte PQ TLS-Unterstützung in Version 3.5. Die einfachste Möglichkeit, diese SDKs und Tools für PQ TLS zu konfigurieren, besteht darin, sie auf einer Betriebssystemdistribution zu verwenden, auf der mindestens OpenSSL 3.5 installiert ist.

Sie können einen Docker-Container auch so konfigurieren, dass er OpenSSL 3.5 verwendet, um PQ TLS auf jedem System zu aktivieren, das Docker unterstützt. Ein Beispiel für die Einrichtung für Python finden Sie unter Post-quantum TLS in Python.

AWS CLI

Ab Version 2.34.54 bündelt das AWS CLI-Installationsprogramm für Linux OpenSSL 3.5.6, sodass PQ TLS standardmäßig für diese und nachfolgende Versionen unter Linux aktiviert und bevorzugt wird. AWS CLI-Benutzer unter Linux können PQ TLS aktivieren, indem sie auf AWS CLI v2.34.54 oder höher aktualisieren.

Installieren Sie für macOS die AWS CLI über Homebrew und stellen Sie sicher, dass Ihr Homebrew-vended OpenSSL auf Version 3.5+ aktualisiert wurde. Sie können dies mit „brew install openssl @3 .6“ tun und mit „brew list | grep openssl“ validieren.

Eine schrittweise Anleitung zur Validierung der Installation finden Sie im Github-Repository und im dazugehörigen Blogbeitrag.

AWS SDK für PHP

Das AWS SDK for PHP ist systemabhängig libssl/libcrypto. Um PQ TLS zu verwenden, verwenden Sie dieses SDK auf einer Betriebssystemdistribution, auf der mindestens OpenSSL 3.5 installiert ist.

AWS SDK für Python (Boto3)

Das AWS SDK for Python (Boto3) basiert auf der OpenSSL-Bibliothek, die mit Ihrer Python-Installation für TLS verknüpft ist. Das Verhalten unterscheidet sich je nach Plattform:

Windows and macOS (python.org installer)

> Die offiziellen python.org-Installationsprogramme für Windows (.exe) und macOS (.pkg) bündeln ihre eigene OpenSSL-Bibliothek. Ab Python 3.14.6 unterstützt und bevorzugt das mitgelieferte OpenSSL 3.5.7 standardmäßig PQ TLS. Es ist keine zusätzliche Konfiguration erforderlich.

macOS (Homebrew)

Da Homebrews Python @3 .14 gegen die gemeinsame Bibliothek openssl @3 von Homebrew verlinkt, wird ab Homebrew OpenSSL 3.5+ PQ TLS standardmäßig unterstützt und bevorzugt. Sie können Ihre OpenSSL-Version validieren mit:

```
python3.14 -c "import ssl; print(ssl.OPENSSL_VERSION)"
```

Wenn Ihre Version unter 3.5 liegt, führen Sie ein Upgrade mit brew upgrade openssl @3 durch.

Linux

Unter Linux verknüpft sich Python dynamisch mit der gemeinsam genutzten libssl des Systems. Die Installation von OpenSSL 3.5+ ist notwendig, aber nicht ausreichend, da die kryptografische Richtlinie des Systems auch Post-Quantum-Gruppen in der TLS-Standardkonfiguration enthalten muss.

Für Amazon Linux 2023 (AL2023.12 oder höher) aktivieren Sie PQ TLS mit:

```
sudo update-crypto-policies --set DEFAULT:PQ
```

Weitere Informationen finden Sie unter Enable Post-Quantum Cryptography (PQC) auf AL2023.

Informationen zu anderen Linux-Distributionen finden Sie in der Dokumentation Ihrer Distribution zur Konfiguration von Standard-TLS-Gruppen.

Sie können überprüfen, ob PQ TLS funktioniert, indem Sie in Ihrem TLS-Handshake nach dem Schlüsselaustausch X25519MLKEM768 suchen.

AWS SDK für Ruby

Das AWS SDK for Ruby ist systemabhängig libssl/libcrypto. Um PQ TLS zu verwenden, verwenden Sie dieses SDK auf einer Betriebssystemdistribution, auf der mindestens OpenSSL 3.5 installiert ist.

AWS SDK für .NET

Unter Linux ist das AWS SDK for .NET vom System abhängig libssl/libcrypto. Um PQ TLS zu verwenden, verwenden Sie dieses SDK auf einer Betriebssystemdistribution, auf der mindestens OpenSSL 3.5 installiert ist. Unter Windows und macOS ist PQ TLS ab in.NET 10 und Windows 11 verfügbar. Unter macOS kann die Unterstützung von TLS 1.3 (eine Voraussetzung für PQ TLS) aktiviert werden, indem Sie sich Network.framework wie hier beschrieben bei Apple anmelden. Unter

der Annahme, dass mindestens eine .NET-Version von 10 vorhanden ist, sollte PQ TLS dann aktiviert sein.

AWS SDKs und Tools planen nicht, PQ TLS zu unterstützen

Derzeit ist nicht geplant, die folgenden Sprach-SDKs und Tools zu unterstützen:

- AWS SDK für SAP
- AWS SDK für Swift
- AWS Tools für Windows PowerShell

AWS Common Runtime (CRT) -Bibliotheken

Die AWS Common Runtime (CRT) -Bibliotheken sind eine Basisbibliothek der SDKs. Die CRT ist eine modulare Familie unabhängiger Pakete, die in C geschrieben sind. Jedes Paket bietet eine gute Leistung und minimalen Platzbedarf für verschiedene erforderliche Funktionen. Diese Funktionen sind allen SDKs gemeinsam und sorgen für eine bessere Wiederverwendung, Optimierung und Genauigkeit von Code. Die Pakete sind:

- `awslabs/aws-c-auth`: AWS clientseitige Authentifizierung (Standardanbieter für Anmeldeinformationen und Signierung (sigv4))
- `awslabs/aws-c-cal`: Primitive kryptografische Typen, Hashes (MD5, SHA256, SHA256 HMAC), Unterzeichner, AES
- `awslabs/aws-c-common`: Grundlegende Datenstrukturen, threading/synchronization primitive Typen, Pufferverwaltung, stdlib-bezogene Funktionen
- `awslabs/aws-c-compression`: Komprimierungsalgorithmen (Huffman) encoding/decoding
- `awslabs/aws-c-event-stream`: Verarbeitung von Event-Stream-Nachrichten (Header, Prelude, Payload), Implementierung von Remote Procedure Call (RPC crc/trailer) über Event-Streams
- `awslabs/aws-c-http`: C99-Implementierung der AND-Spezifikationen HTTP/1.1 HTTP/2
- `awslabs/aws-c-io`: Sockets (TCP, UDP), DNS, Pipes, Ereignisschleifen, Kanäle, SSL/TLS
- `awslabs/aws-c-iot`: C99-Implementierung der Integration von AWS IoT-Cloud-Diensten mit Geräten
- `awslabs/aws-c-mqtt`: Standardmäßiges, leichtes Messaging-Protokoll für das Internet der Dinge (IoT)
- `awslabs/aws-c-s3`: C99-Bibliotheksimplementierung für die Kommunikation mit dem Amazon S3 S3-Service, konzipiert für die Maximierung des Durchsatzes auf Amazon EC2 EC2-Instances mit hoher Bandbreite
- `awslabs/aws-c-sdkutils`: Eine Dienstprogrammbibliothek zum Analysieren und Verwalten von Profilen AWS
- `awslabs/aws-checksums`: Cross-platform hardwarebeschleunigte CRC32c und CRC32 mit Rückgriff auf effiziente Softwareimplementierungen
- `awslabs/aws-lc`: General-purpose kryptografische Bibliothek, die vom AWS Cryptography-Team AWS und seinen Kunden verwaltet wird und auf Code aus dem Google BoringSSL-Projekt und dem OpenSSL-Projekt basiert

- awslabs/s2n: C99-Implementierung der TLS/SSL Protokolle, die so konzipiert sind, dass sie klein und schnell sind, wobei Sicherheit im Vordergrund steht

Das CRT ist in allen SDKs außer Go und Rust verfügbar.

CRT-Abhängigkeiten

Die CRT-Bibliotheken bilden ein komplexes Netz von Beziehungen und Abhängigkeiten. Die Kenntnis dieser Beziehungen ist hilfreich, wenn Sie die CRT direkt aus dem Quellcode erstellen müssen. Die meisten Benutzer greifen jedoch über ihr Sprach-SDK (wie SDK for C++ oder AWS AWS SDK for Java) oder ihr Sprach-IoT-Geräte-SDK (wie IoT SDK for C++ oder AWS AWS IoT SDK for Java) auf CRT-Funktionen zu. In der folgenden Abbildung bezieht sich das Feld Sprach-CRT-Bindungen auf das Paket, das die CRT-Bibliotheken für ein bestimmtes Sprach-SDK umschließt. Dies ist eine Sammlung von Paketen in der Form`aws-crt-*`, wobei '*' für eine SDK-Sprache steht (z. B. aws-crt-cppoder). aws-crt-java

Im Folgenden werden die hierarchischen Abhängigkeiten der CRT-Bibliotheken veranschaulicht. Das CRT-Abhängigkeitsdiagramm zeigt, wie die einzelnen CRT-Bibliotheken miteinander in Beziehung stehen.

AWS Wartungsrichtlinie für SDKs und Tools

-Übersicht

In diesem Dokument werden die Wartungsrichtlinien für AWS Software Development Kits (SDKs) und Tools, einschließlich Mobile- und IoT-SDKs, sowie die zugrunde liegenden Abhängigkeiten beschrieben. AWS versorgt die AWS SDKs und Tools regelmäßig mit Updates, die Unterstützung für neue oder aktualisierte AWS APIs, neue Funktionen, Verbesserungen, Bugfixes, Sicherheitspatches oder Dokumentationsupdates beinhalten können. Updates können sich auch auf Änderungen in Bezug auf Abhängigkeiten, Sprachlaufzeiten und Betriebssysteme beziehen. AWS SDK-Versionen werden für Paketmanager (z. B. Maven NuGet, PyPI) veröffentlicht und sind als Quellcode verfügbar. GitHub

Wir empfehlen Benutzern, sich über SDK-Versionen auf dem Laufenden zu halten, um über die neuesten Funktionen, Sicherheitsupdates und die zugrunde liegenden Abhängigkeiten auf dem Laufenden zu bleiben. Die fortgesetzte Verwendung einer nicht unterstützten SDK-Version wird nicht empfohlen und erfolgt nach eigenem Ermessen des Benutzers.

Versionsverwaltung

Die AWS SDK-Release-Versionen haben die Form, X.Y.Z in der X für die Hauptversion steht.
Die Erhöhung der Hauptversion eines SDK deutet darauf hin, dass dieses SDK erheblichen und wesentlichen Änderungen unterzogen wurde, um neue Redewendungen und Muster in der Sprache zu unterstützen. Hauptversionen werden eingeführt, wenn sich öffentliche Schnittstellen (z. B. Klassen, Methoden, Typen usw.), Verhaltensweisen oder Semantik geändert haben. Anwendungen müssen aktualisiert werden, damit sie mit der neuesten SDK-Version funktionieren. Es ist wichtig, Hauptversionen sorgfältig und gemäß den Upgrade-Richtlinien von zu aktualisieren AWS.

Lebenszyklus der SDK-Hauptversionen

Der Lebenszyklus der wichtigsten SDKs und Tools-Versionen besteht aus 5 Phasen, die im Folgenden beschrieben werden.

- Developer Preview (Phase 0) — In dieser Phase werden SDKs nicht unterstützt, sollten nicht in Produktionsumgebungen verwendet werden und sind nur für den frühen Zugriff und für Feedback vorgesehen. Es ist möglich, dass future Versionen bahnbrechende Änderungen einführen.

Sobald AWS festgestellt wurde, dass es sich bei einer Version um ein stabiles Produkt handelt, kann sie als Release Candidate gekennzeichnet werden. Release Candidates sind bereit für die Veröffentlichung der allgemeinen Version, sofern keine wesentlichen Fehler auftreten, und erhalten vollen AWS Support.

- Allgemeine Verfügbarkeit (GA) (Phase 1) — In dieser Phase werden SDKs vollständig unterstützt. AWS wird regelmäßige SDK-Versionen bereitstellen, die Unterstützung für neue Dienste, API-Updates für bestehende Dienste sowie Fehler- und Sicherheitskorrekturen beinhalten. For Tools AWS wird regelmäßige Versionen bereitstellen, die neue Funktionsupdates und Bugfixes enthalten. AWS unterstützt die GA-Version eines SDK mindestens 24 Monate lang.
- Wartungsankündigung (Phase 2) — AWS Eine öffentliche Ankündigung erfolgt mindestens 6 Monate, bevor ein SDK in den Wartungsmodus wechselt. Während dieses Zeitraums wird das SDK weiterhin vollständig unterstützt. In der Regel wird der Wartungsmodus gleichzeitig mit der Umstellung der nächsten Hauptversion auf GA angekündigt.
- Wartung (Phase 3) — AWS Beschränkt SDK-Versionen während des Wartungsmodus auf kritische Bugfixes und Sicherheitsprobleme. Ein SDK erhält keine API-Updates für neue oder bestehende Dienste und wird auch nicht aktualisiert, um neue Regionen zu unterstützen. Der Wartungsmodus hat eine Standarddauer von 12 Monaten, sofern nicht anders angegeben.
- End-of-Support (Phase 4) — Wenn der Support für ein SDK ausläuft, erhält es keine Updates oder Releases mehr. Zuvor veröffentlichte Versionen werden weiterhin über öffentliche Paketmanager verfügbar sein und der Code bleibt aktiviert GitHub. Das GitHub Repository kann archiviert werden. Die Verwendung eines SDK, für das der Support abgelaufen ist, liegt im Ermessen des Benutzers. Wir empfehlen Benutzern, auf die neue Hauptversion zu aktualisieren.

Im Folgenden finden Sie eine visuelle Darstellung des Lebenszyklus der SDK-Hauptversion. Bitte beachten Sie, dass die unten angegebenen Zeitpläne der Veranschaulichung dienen und nicht bindend sind.
Zeitpläne für die Wartungsrichtlinien

Lebenszyklus von Abhängigkeiten

Den meisten AWS SDKs liegen Abhängigkeiten zugrunde, wie z. B. Sprachlaufzeiten, Betriebssysteme oder Bibliotheken und Frameworks von Drittanbietern. Diese Abhängigkeiten sind in der Regel an die Sprachgemeinschaft oder den Anbieter gebunden, dem die jeweilige Komponente gehört. Jede Community oder jeder Anbieter veröffentlicht ihren eigenen Zeitplan für das Ende des Supports für ihr Produkt.

Die folgenden Begriffe werden verwendet, um die zugrunde liegenden Abhängigkeiten von Drittanbietern zu klassifizieren:

- Betriebssystem (OS): Beispiele hierfür sind Amazon Linux AMI, Amazon Linux 2, Windows 2008, Windows 2012, Windows 2016 usw.
- Language Runtime: Zu den Beispielen gehören Java 7, Java 8, Java 11, .NET Core, .NET Standard, .NET PCL usw.
- Bibliothek eines Drittanbieters//Framework: Beispiele hierfür sind OpenSSL, .NET Framework 4.5, Java EE usw.

Unsere Richtlinie sieht vor, SDK-Abhängigkeiten noch mindestens 6 Monate lang zu unterstützen, nachdem die Community oder der Anbieter den Support für die Abhängigkeit eingestellt hat. Diese Richtlinie kann jedoch je nach spezifischer Abhängigkeit variieren.

> **Note**
>
> AWS behält sich das Recht vor, den Support für eine zugrunde liegende Abhängigkeit einzustellen, ohne die SDK-Hauptversion zu erhöhen

Methoden der Kommunikation

Wartungsankündigungen werden auf verschiedene Arten kommuniziert:

- An die betroffenen Konten wird eine E-Mail-Benachrichtigung gesendet, in der unsere Pläne angekündigt werden, den Support für die jeweilige SDK-Version einzustellen. In der E-Mail wird der Weg bis zum Ende des Supports beschrieben, der Zeitplan für die Kampagne angegeben und Hinweise zum Upgrade gegeben.
- AWS Die SDK-Dokumentation, z. B. API-Referenzdokumentation, Benutzerhandbücher, SDK-Produktmarketingseiten und GitHub Readme-Dateien, wurden aktualisiert, um den Zeitplan der Kampagne anzugeben und Hinweise zur Aktualisierung der betroffenen Anwendungen zu geben.
- Es wird ein AWS Blogbeitrag veröffentlicht, der den Weg bis zum Ende des Supports beschreibt und die Zeitpläne der Kampagne wiederholt.
- Den SDKs wurden Warnmeldungen hinzugefügt, die den Weg bis zum Ende des Supports beschreiben und auf die SDK-Dokumentation verweisen.

Eine Liste der verfügbaren Hauptversionen von AWS SDKs und Tools sowie deren Status im Wartungszyklus finden Sie unter. Lebenszyklus der Version

AWS Versionslebenszyklus von SDKs und Tools

Die folgende Tabelle zeigt die Liste der verfügbaren Hauptversionen des AWS Software Development Kit (SDK) und deren Position im Wartungslebenszyklus mit den zugehörigen Zeitplänen. Detaillierte Informationen zum Lebenszyklus der Hauptversionen von AWS SDKs und Tools und den zugrunde liegenden Abhängigkeiten finden Sie unter. Wartungsrichtlinie

SDK	Hauptversion	Aktuelle Phase	Datum der allgemeinen Verfügbarkeit	Hinweise
AWS CLI	1.x	Ankündigung der Wartung	9/2/2013	Einzelheiten und Termine finden Sie in der Ankündigung
AWS CLI	2.x	Allgemeine Verfügbarkeit	2/10/2020	
SDK for C++	1.x	Allgemeine Verfügbarkeit	9/2/2015	
SDK for Go V2	V2 1.x	Allgemeine Verfügbarkeit	1/19/2021	
SDK for Go	1.x	End-of-Support	11/19/2015	
SDK für Java	1.x	End-of-Support	3/25/2010	
SDK für Java	2.x	Allgemeine Verfügbarkeit	11/20/2018	
SDK für JavaScript	1.x	End-of-Support	5/6/2013	
SDK für JavaScript	2.x	End-of-Support	6/19/2014	

SDK	Hauptversion	Aktuelle Phase	Datum der allgemeinen Verfügbarkeit	Hinweise
SDK für JavaScript	3.x	Allgemeine Verfügbarkeit	12/15/2020	
SDK für Kotlin	1.x	Allgemeine Verfügbarkeit	11/27/2023	
SDK for .NET	1.x	End-of-Support	11/2009	
SDK for .NET	2.x	End-of-Support	11/8/2013	
SDK for .NET	3.x	End-of-Support	6/01/2026	
SDK for .NET	4.x	Allgemeine Verfügbarkeit	4/28/2025	
SDK for PHP	2.x	End-of-Support	11/2/2012	
SDK for PHP	3.x	Allgemeine Verfügbarkeit	5/27/2015	
SDK für Python (Boto2)	1.x	End-of-Support	7/13/2011	
SDK for Python (Boto3)	1.x	Allgemeine Verfügbarkeit	6/22/2015	
SDK für Python (Botocore)	1.x	Allgemeine Verfügbarkeit	6/22/2015	
SDK for Ruby	1.x	End-of-Support	7/14/2011	
SDK for Ruby	2.x	End-of-Support	2/15/2015	
SDK for Ruby	3.x	Allgemeine Verfügbarkeit	8/29/2017	

SDK	Hauptversion	Aktuelle Phase	Datum der allgemeinen Verfügbarkeit	Hinweise
SDK für Rust	1.x	Allgemeine Verfügbarkeit	11/27/2023	
SDK für Swift	1.x	Allgemeine Verfügbarkeit	9/17/2024	
Werkzeuge für PowerShell	2.x	End-of-Support	11/8/2013	
Tools für PowerShell	3.x	End-of-Support	7/29/2015	
Tools für PowerShell	4.x	End-of-support	06/01/2026	
Werkzeuge für PowerShell	5.x	Allgemeine Verfügbarkeit	6/23/2025	

Suchen Sie nach einem SDK oder Tool, das nicht erwähnt wird? Verschlüsselungs-SDKs, IoT-Geräte-SDKs und mobile SDKs sind beispielsweise nicht in diesem Handbuch enthalten. Dokumentation zu diesen anderen Tools finden Sie unter Tools, auf denen Sie aufbauen können. AWS

Dokumentenverlauf für AWS SDKs und Referenzhandbuch zu Tools

In der folgenden Tabelle werden wichtige Ergänzungen und Aktualisierungen des Referenzhandbuchs AWS SDKs und der Tools beschrieben. Um Benachrichtigungen über Aktualisierungen dieser Dokumentation zu erhalten, können Sie den RSS-Feed abonnieren.

Änderung	Beschreibung	Datum
Neue S3 Express One Zone-Einstellung wird hinzugefügt	Neue S3 Express One Zone-Einstellung zur Deaktivierung der Sitzungsauthentifizierung hinzugefügt.	13. Oktober 2025
Eine neue Entscheidungsstruktur für die Authentifizierung wird hinzugefügt	Es wurde ein neuer Entscheidungsbaum hinzugefügt, der bei Authentifizierungsentscheidungen zwischen den Optionen hilft.	23. September 2025
Eine neue Funktion für das Authentifizierungsschema wird hinzugefügt	Neue Funktion für das Authentifizierungsschema wird hinzugefügt. Updates für AWS STS regionale Endpunkte.	18. August 2025
Neue Version von Tools für wird hinzugefügt PowerShell	Die neueste Version von Tools for PowerShell Support wird zu allen Einstellungen hinzugefügt. Kompatibilität mit AWS SDKs Tabellen. Funktion zur Injektion von Hostpräfixen hinzugefügt.	23. Juni 2025
Aktualisierungen des Seitentitels	Weitere Titel, Tabellentitel, Zusammenfassungen und SEO-Updates.	5. März 2025

Aktualisierungen der Seitentitel	Der Inhalt wird aktualisiert, um aussagekräftigere Titel zu verwenden.	24. Februar 2025
Swift SDK zur Einstellungsreferenz hinzufügen	Swift SDK-Unterstützung zu allen Einstellungsreferenzen hinzufügen Kompatibilität mit AWS SDKs Tabellen.	17. September 2024
SDK for Java 1.x-Systemeigenschaften	Fügen Sie Details zu den unterstützten JVM-Systemkonfigurationseinstellungen von Version 1.x hinzu. AWS SDK für Java	30. Mai 2024
Aktualisierungen der Einstellungen	Fügen Sie JVM-Systemkonfigurationseinstellungen hinzu.	27. März 2024
Aktualisierungen der Kompatibilitätstabelle	Aktualisierungen der Kompatibilität für die SDK-Unterstützung, Aktualisierungen der IAM Identity Center-Verfahren.	20. Februar 2024
Aktualisierung der Container-Anmeldeinformationen. IMDS-Aktualisierung.	Unterstützung für Amazon EKS wird hinzugefügt. Einstellung zur Deaktivierung von IMDSv1 Fallback hinzugefügt.	29. Dezember 2023
Komprimierung anfordern	Hinzufügen von Einstellungen für die Funktion zur Komprimierung von Anfragen	27. Dezember 2023

Kompatibilitätstabellen	Die Kompatibilitätstabellen für SDK- und Toolfunktionen wurden aktualisiert und enthalten nun SDK für Kotlin, SDK für Rust und AWS -Tools für PowerShell.	10. Dezember 2023
Aktualisierungen der Authentifizierung	Aktualisierungen der unterstützten Authentifizierungsmethoden SDKs und Tools.	1. Juli 2023
Aktualisierungen der bewährten Methoden für IAM	Aktualisierung des Leitfadens zur Ausrichtung an bewährten IAM-Methoden. Weitere Informationen finden Sie unter Bewährte Sicherheitsmethoden in IAM.	27. Februar 2023
SSO-Aktualisierungen	Aktualisierungen der SSO-Anmeldeinformationen für die neue SSO-Token-Konfiguration.	19. November 2022
Aktualisierungen der Einstellungen	Aktualisierungen der Unterstützungstabelle für die allgemeine Konfiguration und für Amazon S3 Multi-Region Access Points.	17. November 2022
Aktualisierungen der Einstellungen	Aktualisierungen zur besseren Übersicht der IMDS-Client- und IMDS-Anmeldeinformationen. Aktualisierungen der Umgebungsvariablen.	04. November 2022
Die Willkommensseite wird aktualisiert	Ankündigung von Amazon CodeWhisperer.	22. September 2022

Änderung des Dienstnamens für Single Sign-On	Aktualisierungen, die dem Umstand Rechnung tragen, dass AWS SSO jetzt als AWS IAM Identity Center bezeichnet wird.	26. Juli 2022
Die Einstellungen werden aktualisiert	Kleinere Aktualisierungen der Details der Konfigurationsdatei und der unterstützten Einstellungen.	15. Juni 2022
Aktualisieren	Umfangreiches Update fast aller Teile dieses Handbuchs.	1. Februar 2022
Erstversion	Die erste Version dieses Handbuchs wurde der Öffentlichkeit zugänglich gemacht.	13. März 2020

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.