



Referenzhandbuch

AWS SDKs und Tools



AWS SDKs und Tools: Referenzhandbuch

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

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

Table of Contents

AWS SDKsReferenzhandbuch für Tools und Tools	1
Ressourcen für Entwickler	2
Telemetrie-Benachrichtigung im Toolkit	3
Konfiguration	4
Geteilte credentials Dateien config und Dateien	5
Profile	5
Format der Konfigurationsdatei	7
Format der Datei mit den Anmeldeinformationen	10
Speicherort der gemeinsam genutzten Dateien	11
Auflösung des Home-Verzeichnisses	12
Ändern Sie den Standardspeicherort dieser Dateien	12
Umgebungsvariablen	14
Festlegen von Umgebungsvariablen	14
Einrichtung von serverlosen Umgebungsvariablen	15
JVM-Systemeigenschaften	16
Wie legt man die JVM-Systemeigenschaften fest	16
Authentifizierung und Zugriff	19
AWS Builder ID	21
IAMIdentity Center-Authentifizierung	21
Konfigurieren Sie den programmatischen Zugriff mithilfe von IAM Identity Center	22
Verstehen Sie die IAM Identity Center-Authentifizierung	25
IAM Roles Anywhere	30
Schritt 1: Konfigurieren Sie IAM Roles Anywhere	30
Schritt 2: Verwenden Sie IAM Roles Anywhere	31
Übernehmen einer Rolle	32
Nehmen Sie eine Rolle an IAM	33
Nehmen Sie eine Rolle an (Web)	34
Verbunden mit Web-Identität oder OpenID Connect	35
AWS -Zugriffsschlüssel	37
Verwenden kurzfristiger Anmeldeinformationen	37
Verwenden langfristiger Anmeldeinformationen	37
Kurzfristige Anmeldeinformationen	38
Langfristige Anmeldeinformationen	40
IAMRollen für EC2 Instanzen	44

Erstellen Sie eine IAM-Rolle	44
Starten Sie eine EC2 Amazon-Instance und geben Sie Ihre IAM Rolle an	45
Connect zur EC2 Instanz her	45
Führen Sie Ihre Anwendung auf der Instanz aus EC2	46
Referenz zu Einstellungen	47
Serviceclients erstellen	47
Vorrang der Einstellungen	47
Seiten mit Einstellungen	49
ConfigListe der Dateieinstellungen	50
CredentialsListe der Dateieinstellungen	55
Liste der Umgebungsvariablen	55
Liste der JVM-Systemeigenschaften	60
Standardisierte Anbieter von Anmeldeinformationen	63
Verstehen Sie die Kette der Anbieter von Anmeldeinformationen	63
SDK-spezifische und toolspezifische Anbieterketten für Anmeldeinformationen	65
AWS Zugriffstasten	66
Nehmen Sie die Rolle des Anbieters an	69
Container-Anbieter	76
IAMIdentity Center-Anbieter	80
IMDSAnbieter	86
Prozessanbieter	91
Standardisierte Funktionen	95
Kontobasierte Endpunkte	96
Application ID	99
EC2Amazon-Instanz-Metadaten	101
Amazon S3 Access Points	103
Multiregionale Amazon-S3-Zugriffspunkte	105
AWS-Region	107
AWS STS Regionale Endpunkte	111
Schutz der Datenintegrität	115
Dual-Stack und Endpunkte FIPS	120
Endpunkterkennung	123
Allgemeine Konfiguration	125
IMDSKlient	128
Wiederholungsverhalten	131
Komprimierung anfordern	138

Servicespezifische Endpunkte	140
Standardeinstellungen für intelligente Konfigurationen	187
Allgemeine Runtime	193
CRTAbhängigkeiten	194
Wartungsrichtlinie	195
Übersicht	195
Versionsverwaltung	195
Lebenszyklus der SDK-Hauptversionen	195
Lebenszyklus von Abhängigkeiten	196
Methoden der Kommunikation	197
Versionsunterstützung	199
Dokumentverlauf	202
AWS-Glossar	205
.....	ccvi

AWS SDKsReferenzhandbuch für Tools und Tools

Viele SDKs dieser Tools weisen einige gemeinsame Funktionen auf, entweder durch gemeinsame Konstruktionsspezifikationen oder durch eine gemeinsame Bibliothek.

Dieses Handbuch enthält Informationen zu:

- [Konfiguration](#)— Wie Sie die Variablen `shared config` und `credentials files` oder `environment` verwenden, um Ihre Tools AWS SDKs zu konfigurieren.
- [Authentifizierung und Zugriff](#)— Stellen Sie fest, wie sich Ihr Code oder Tool authentifiziert AWS , wenn Sie mit AWS-Services entwickeln.
- [Referenz zu Einstellungen](#)— Referenz für alle standardisierten Einstellungen, die für die Authentifizierung und Konfiguration verfügbar sind.
- [AWS Allgemeine Runtime \(CRT\) -Bibliotheken](#)— Überblick über die gemeinsam genutzten AWS Common Runtime (CRT) -Bibliotheken, die für fast alle verfügbar sindSDKs.
- [AWS Wartungsrichtlinie für SDKs und Tools](#)behandelt die Wartungsrichtlinien und die Versionierung für AWS Software Development Kits (SDKs) und Tools, einschließlich Mobile und Internet of Things (IoT)SDKs, sowie die zugrunde liegenden Abhängigkeiten.

Dieses Referenzhandbuch AWS SDKs und das Referenzhandbuch für Tools sollen als Informationsbasis für mehrere SDKs Tools dienen. Zusätzlich zu den SDK hier aufgeführten Informationen sollte der spezifische Leitfaden für das von Ihnen verwendete Tool verwendet werden. Im Folgenden finden Sie die Tools SDK und Tools mit entsprechenden Abschnitten in diesem Handbuch:

Wenn Sie Folgendes verwenden:	Die für Sie relevanten Abschnitte dieses Handbuchs sind:
<ul style="list-style-type: none"> • SDKIrgendein Werkzeug 	AWS Wartungsrichtlinie für SDKs und Tools
<ul style="list-style-type: none"> • AWS Cloud9 • AWS CDK • AWS Toolkit for Azure DevOps • AWS Toolkit for JetBrains • AWS Toolkit for Visual Studio 	Konfiguration Authentifizierung und Zugriff AWS Wartungsrichtlinie für SDKs und Tools

Wenn Sie Folgendes verwenden:	Die für Sie relevanten Abschnitte dieses Handbuchs sind:
<ul style="list-style-type: none">• AWS Toolkit for Visual Studio Code• AWS Serverless Application Model • AWS CodeArtifact• AWS CodeBuild• Amazon CodeCatalyst• AWS CodeCommit• AWS CodeDeploy• AWS CodePipeline	
<ul style="list-style-type: none">• AWS CLI• AWS SDK for C++• AWS SDK für Go• AWS SDK for Java• AWS SDK for JavaScript• AWS SDK for Kotlin• AWS SDK for .NET• AWS SDK for PHP• AWS SDK for Python (Boto3)• AWS SDK for Ruby• AWS SDK for Rust• AWS SDK for Swift• AWS Tools for Windows PowerShell	<ul style="list-style-type: none">• Konfiguration• Authentifizierung und Zugriff• Referenz zu Einstellungen• AWS Allgemeine Runtime (CRT) -Bibliotheken• AWS Wartungsrichtlinie für SDKs und Tools• AWS SDKs und Tools-Versionsunterstützung

Ressourcen für Entwickler

Einen Überblick über Tools, mit denen Sie Anwendungen entwickeln können, finden Sie unter [Tools AWS, auf denen Sie aufbauen](#) können AWS. Informationen zum Support finden Sie im [AWS Knowledge Center](#).

Amazon Q Developer ist ein generativer KI-gestützter Konversationsassistent, der Ihnen helfen kann, Anwendungen zu verstehen, zu erstellen, zu erweitern und zu betreiben AWS . 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 Amazon Q Developer User Guide.

Telemetrie-Benachrichtigung im Toolkit

AWS Toolkits für die integrierte Entwicklungsumgebung (IDE) sind Plugins und Erweiterungen, die den Zugriff auf AWS Dienste in Ihrem ermöglichen. IDE Amazon IDE Q-Plugins und -Erweiterungen ermöglichen generative KI-Unterstützung in Ihrem IDE. Detaillierte Informationen zu den einzelnen IDE Toolkits finden Sie in den Toolkit-Benutzerhandbüchern in der obigen Tabelle. Weitere Informationen zur Verwendung von Amazon Q in Ihrem IDE finden Sie im IDE Thema [Verwenden von Amazon Q 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 Dokument [commonDefinitions.json](#) im `aws-toolkit-common` Github-Repository.

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

- [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.

Konfiguration

Mit AWS SDKs und anderen AWS Entwicklertools wie dem AWS Command Line Interface (AWS CLI) können Sie mit AWS Service-APIs interagieren. 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 meisten AWS SDKs und Tools unterstützen jedoch allgemeine Einstellungen aus zwei Hauptquellen (über den Code selbst hinaus):

- [Dateien mit gemeinsam genutzten AWS Konfigurationen und Anmeldeinformationen](#) — Die gemeinsam genutzten `credentials` Dateien `config` und Dateien sind die gängigste Methode, um die Authentifizierung und Konfiguration für ein AWS SDK oder Tool festzulegen. Verwenden Sie diese Dateien, um Einstellungen zu speichern, die Ihre Tools und Anwendungen verwenden können. Die Einstellungen in den geteilten `credentials` Dateien `config` und Dateien sind einem bestimmten Profil zugeordnet. Bei mehreren Profilen können Sie unterschiedliche Einstellungskonfigurationen erstellen, die in verschiedenen Szenarien angewendet werden können. Wenn Sie ein AWS Tool zum Aufrufen eines Befehls oder ein SDK zum Aufrufen einer AWS API verwenden, können Sie angeben, welches Profil und somit welche Konfigurationseinstellungen für diese Aktion verwendet werden sollen. Eines der Profile ist als `default` Profil gekennzeichnet und wird automatisch verwendet, wenn Sie nicht explizit ein zu verwendendes Profil angeben. Die

Einstellungen, die Sie in diesen Dateien speichern können, sind in diesem Referenzhandbuch dokumentiert.

- [Umgebungsvariablen](#) — Einige der Einstellungen können alternativ in den Umgebungsvariablen Ihres Betriebssystems gespeichert werden. Sie können zwar jeweils nur einen Satz von Umgebungsvariablen verwenden, diese können jedoch leicht dynamisch geändert werden, wenn Ihr Programm ausgeführt wird und sich Ihre Anforderungen ändern.

Weitere Themen in diesem Abschnitt

- [Geteilte credentials Dateien config und Dateien](#)
- [Speicherort der geteilten credentials Dateien config und Dateien](#)
- [Unterstützung von Umgebungsvariablen](#)
- [Unterstützung für JVM-Systemeigenschaften](#)

Geteilte **credentials** Dateien **config** und Dateien

Die geteilten `credentials` Dateien `AWS config` und Dateien enthalten eine Reihe von Profilen. Ein Profil ist ein Satz von Konfigurationseinstellungen in Schlüssel-Wert-Paaren, der von den Tools AWS Command Line Interface (AWS CLI) AWS SDKs, dem und anderen verwendet wird. Konfigurationswerte werden an ein Profil angehängt, um einen bestimmten Aspekt des SDK /tools zu konfigurieren, wenn dieses Profil verwendet wird. Diese Dateien werden „gemeinsam genutzt“, da 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 enthalten (UTF-8-kodiert). [Sie haben die Form von Dateien, die allgemein als Dateien bezeichnet werden. INI](#)

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 einer Operation des SDK Oder-Tools verwendet werden, wenn kein bestimmtes benanntes Profil angegeben ist. Sie können auch separate Profile

erstellen, auf die Sie explizit namentlich verweisen können. Jedes Profil kann je nach Anwendung und Szenario unterschiedliche Einstellungen und Werte verwenden.

Note

[default] ist einfach ein unbenanntes Profil. Dieses Profil ist benannt default, weil es das Standardprofil ist, das von verwendet wird, SDK wenn der Benutzer kein Profil angibt. Es stellt keine vererbten Standardwerte für andere Profile bereit. 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 OR-Tool bei der Ausführung Ihres Codes verwendet werden. Profile können auch innerhalb des Codes oder per 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

SDKFür Kotlin auf dem JVM und SDK für Java 2.x können Sie [die aws.profile Systemeigenschaft setzen](#). Wenn der einen Service-Client SDK erstellt, verwendet er die

Einstellungen im genannten Profil, sofern die Einstellung nicht im Code überschrieben wird. Das SDK für 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 unter [IAM Identity Center-Authentifizierung für Ihr Tool SDK oder](#). 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 Sie verfügbar ist SDK, finden Sie unter Servicespezifische [Kompatibilität mit AWS SDKs](#) Endgeräte.

`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.

Im folgenden Beispiel wird eine `services` Definition verwendet, 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 lautet `dynamodb`. Der Unterabschnitt Amazon DynamoDB Service beginnt in der Zeile `dynamodb =` . 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 unter [Servicespezifische 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 `profile` beginnen. 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_id`, `aws_secret_access_key`, `aws_session_token`. Diese Einstellungen können zwar auch in der gemeinsam

Betriebssystem	Standardspeicherort und Name der Dateien
	<code>~/.aws/credentials</code>
Windows	<code>%USERPROFILE%\aws\config</code> <code>%USERPROFILE%\aws\credentials</code>

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 werden ~\ beide 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:

- configDatei-Umgebungsvariable: **AWS_CONFIG_FILE**
- credentialsDatei-Umgebungsvariable: **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 unter [Unterstützung von Umgebungsvariablen](#).

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:

- configDatei-JVM-Systemeigenschaft: **aws.configFile**
- credentialsDatei-Umgebungsvariable: **aws.sharedCredentialsFile**

Anweisungen zum Einstellen der JVM-Systemeigenschaften finden Sie unter [the section called “Wie legt man die JVM-Systemeigenschaften fest”](#). Das SDK for Java 1.x unterstützt diese Systemeigenschaften nicht.

Unterstützung von Umgebungsvariablen

Umgebungsvariablen sind eine weitere Möglichkeit, Konfigurationsoptionen und Anmeldeinformationen anzugeben. Sie sind nützlich, wenn Sie Skripts erstellen oder vorübergehend ein benanntes Profil als Standard festlegen möchten. Eine Liste der Umgebungsvariablen, die von den meisten unterstützt werden SDKs, finden Sie unter [Liste 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...zrkuWJ0gQs8IZZaIv2BXIa2R40lgk
$ 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 wJa1rXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY
C:\> setx
  AWS_SESSION_TOKEN AQoEXAMPLEH4aoAH0gNCAPy...truncated...zrkuWJ0gQs8IZZaIv2BXIa2R40Lgk
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="wJa1rXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY"
PS C:\>
PS C:\> $Env:AWS_SESSION_TOKEN="AQoEXAMPLEH4aoAH0gNCAPy...truncated...zrkuWJ0gQs8IZZaIv2BXIa2R40Lgk"
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 zu 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.

Unterstützung für JVM-Systemeigenschaften

[JVM-Systemeigenschaften](#) bieten eine weitere Möglichkeit, Konfigurationsoptionen und Anmeldeinformationen für SDKs anzugeben, die auf der JVM ausgeführt werden, wie z. B. der und der. AWS SDK for Java AWS SDK for Kotlin [Eine Liste der von SDKs unterstützten JVM-Systemeigenschaften 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 in Situationen nützlich, z. B. beim Ausführen einer AWS Lambda Funktion in der Java-Laufzeit oder beim Ausführen von Code in einer eingebetteten JVM.

Das folgende Beispiel konfiguriert AWS-Region global für alle Service-Clients, 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

Sie müssen bei der Entwicklung mit festlegen, wie Ihr Code authentifiziert AWS wird. AWS-Services Sie können den programmgesteuerten Zugriff auf AWS Ressourcen je nach Umgebung und verfügbarem AWS Zugriff auf unterschiedliche Weise konfigurieren.

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

- [IAM Identity Center-Authentifizierung für Ihr Tool SDK oder](#)— Aus Sicherheitsgründen empfehlen wir die Verwendung AWS Organizations zusammen mit IAM Identity Center, um den Zugriff für alle 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
- [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.
- [Nehmen Sie eine Rolle mit AWS Anmeldeinformationen an](#)— Sie können eine IAM-Rolle annehmen, um vorübergehend auf AWS Ressourcen zuzugreifen, auf die Sie sonst möglicherweise keinen Zugriff hätten.
- [AWS -Zugriffsschlüssel](#)— Andere Optionen, die möglicherweise weniger praktisch sind oder das Sicherheitsrisiko für Ihre AWS Ressourcen erhöhen könnten.

Authentifizierungsoptionen 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.

- [IAM-Rollen für EC2 Amazon-Instances verwenden](#)— Verwenden Sie IAM-Rollen, um Ihre Anwendung sicher auf einer EC2 Amazon-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 unter [Nehmen Sie eine Rolle mit Web-Identität oder OpenID Connect an](#).

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.

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-Referenz enthält 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.

AWS Builder ID

Ihre AWS Builder ID Ergänzung zu allen Produkten, die AWS-Konten Sie vielleicht bereits 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.
- [CodeCatalyst Konzepte — AWS Builder ID](#) im CodeCatalyst Amazon-Benutzerhandbuch — Erfahren Sie, wie ein CodeCatalyst verwendet wird AWS Builder ID.

IAM Identity Center-Authentifizierung für Ihr Tool SDK oder

AWS IAM Identity Center ist die empfohlene Methode zur Bereitstellung von AWS Anmeldeinformationen bei der Entwicklung auf einem Dienst ohne AWS Rechenleistung. Das wäre zum Beispiel so etwas wie Ihre lokale Entwicklungsumgebung. Wenn Sie auf einer AWS Ressource wie Amazon Elastic Compute Cloud (AmazonEC2) oder entwickeln, empfehlen wir AWS Cloud9, stattdessen Anmeldeinformationen von diesem Service zu beziehen.

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 Zugriffportal ist die Webadresse, über die Sie sich manuell beim IAM Identity Center anmelden. Das Format von URL ist `d-xxxxxxxxx.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.

Dieses Verfahren verwendet das AWS Zugriffsportal, um Konfigurationswerte abzurufen, die Sie für den SDK /tool-Authentifizierungsprozess benötigen.

- Das AWS CLI wird verwendet, um Ihr SDK Tool so zu konfigurieren, dass es die IAM Identity Center-Authentifizierung für API Anrufe verwendet, die mit Ihrem 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.

Konfigurieren Sie den programmatischen Zugriff mithilfe von IAM Identity Center

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

Wenn Sie IAM Identity Center noch nicht aktiviert haben, finden Sie [weitere Informationen unter IAM Identity Center aktivieren](#) im AWS IAM Identity Center Benutzerhandbuch.

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 `PowerUserAccess`, wenn Sie mit Ihrem Tool arbeiten/SDK.

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 Nutzung 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 Ihnen dies zur Verfügung stellen URL oder Sie erhalten es in einer E-Mail nach Schritt 1: Zugang einrichten. Wenn nicht, finden Sie Ihr AWS Zugangsportal URL 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 einen beliebigen Berechtigungssatz, den Sie oder Ihr Arbeitgeber erstellt haben, 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-Anmeldeinformationsmethode, um die `SSO Region` Werte `Issuer URL` und 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 können, finden Sie unter [Profile](#).
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 IAM Identity Center-Berechtigungssatz angezeigt werden, den Sie konfiguriert haben.

5. Wenn Sie eine verwenden AWS SDK, erstellen Sie eine Anwendung für Sie SDK in Ihrer Entwicklungsumgebung.
 - a. Bei einigen SDKs SS00IDC 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 zuvor den Zugriff auf konfiguriert haben AWS, überprüfen Sie Ihre geteilte `AWS credentials` Datei auf etwaige [AWS Zugriffstasten](#). Aufgrund der [Verstehen Sie die Kette der Anbieter von Anmeldeinformationen](#) Rangfolge müssen Sie alle statischen Anmeldeinformationen entfernen, bevor das SDK Oder-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 [Verstehen Sie die IAM Identity Center-Authentifizierung](#).

Abhängig von der Länge Ihrer konfigurierten Sitzung läuft Ihr Zugriff irgendwann ab und beim Tool SDK oder tritt ein Authentifizierungsfehler auf. 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 AWS CLI anmelden müssen. Weitere Informationen finden Sie in folgenden Themen im AWS IAM Identity Center -Benutzerhandbuch:

- IAM Identity Center-Sitzungsdauer — [Konfigurieren Sie die Dauer der AWS Zugriffsportalsitzungen Ihrer Benutzer](#)
- Sitzungsdauer per Berechtigungssatz — [Sitzungsdauer](#) festlegen

Einzelheiten zu allen IAM Identity Center-Anbiereinstellungen SDKs und Tools finden Sie [IAM Identity Center-Anmeldeinformationsanbieter](#) in diesem Handbuch.

Verstehen Sie die IAM Identity Center-Authentifizierung

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 SDK-APIs werden für einige dieser Authentifizierungskonzepte andere Namen als für IAM Identity Center 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	Beschreibung
Identitätszentrum	sso	Obwohl AWS Single Sign-On umbenannt wurde, behalten die sso API-Namespace aus Gründen der Abwärtskompatibilität ihren ursprünglichen Namen. Weitere Informationen finden Sie unter Umbenennung von IAM Identity Center im Benutzerhandbuch . AWS IAM Identity Center
IAM Identity Center-Konsole Administrationskonsole		Die Konsole, mit der Sie Single Sign-On konfigurieren.
AWSauf die Portal-URL zugreifen		Eine eindeutige URL für Ihr IAM Identity Center-Konto, wie <code>https://xxx.awsapps.com/start</code> . 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.

Name des IAM Identity Center	SDK-API-Name	Beschreibung
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.
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. Bei den meisten AWS-Services handelt es sich um 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 einer AWS-Zugriffsportalsitzung

- 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 Aktualisierungstoken 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 bieten Zugriff darauf, AWS-Services wenn Sie ein Client-Objekt pro Dienst instanziiieren. 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 zu starten. 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 [getRoleCredentials](#) der 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 [SSO-Konfiguration des 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-Zugriffssportalsitzung nicht abgelaufen ist.

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: Konfigurieren Sie IAM Roles Anywhere

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-Profiles 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 Die SDKs und AWS CLI unterstützen einen Prozessanmeldedienstanbieter, der zur Authentifizierung verwendet wird. `credential_process` 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 Management Console. 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 Prozessanmeldedaten](#) in diesem Handbuch.

Nehmen Sie eine Rolle mit AWS Anmeldeinformationen an

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 APIReferenz unter [Aktionen](#).

Um Ihr SDK Tool für die Übernahme einer Rolle einzurichten, müssen Sie zunächst eine bestimmte Rolle erstellen oder identifizieren, die Sie übernehmen möchten. IAMRollen werden eindeutig durch eine Rolle identifiziert Amazon Resource Name ([ARN](#)). 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 IAM Rollen verwenden](#) im IAMBenutzerhandbuch.

Nachdem die IAM Rolle identifiziert wurde und Sie diese Rolle als vertrauenswürdig einstufen, können Sie Ihr SDK Tool so konfigurieren, dass es die von der Rolle gewährten Berechtigungen verwendet.

Note

Es hat sich AWS bewährt, wann immer möglich regionale Endpunkte zu verwenden und Ihre [AWS-Region](#) zu konfigurieren.

Nehmen Sie eine Rolle an IAM

Wenn Sie eine Rolle übernehmen, wird ein Satz temporärer Sicherheitsanmeldedaten AWS STS zurückgegeben. 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 Tool für die Übernahme einer Rolle einzurichten, müssen Sie zunächst eine bestimmte Rolle erstellen oder identifizieren, die Sie übernehmen möchten. IAMRollen werden anhand einer Rolle eindeutig identifiziert [ARN](#). Rollen stellen Vertrauensbeziehungen zu einer anderen Entität her, in der Regel innerhalb Ihres Kontos oder für kontoübergreifenden Zugriff. Informationen zur Einrichtung finden Sie unter [IAMRollen erstellen](#) im IAMBenutzerhandbuch.

Schritt 2: Konfigurieren Sie das SDK Oder-Tool

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

Wird verwendet `credential_source`, um Anmeldeinformationen aus einem ECS Amazon-Container, einer EC2 Amazon-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 `source_profile` oder `credential_source`
- (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
source_profile = profile-name-with-user-that-can-assume-role
```

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

Einzelheiten zu allen Einstellungen des Anbieters für die Übernahme der Rollenmeldedaten finden Sie [Übernehmen Sie die Rolle Credential Provider](#) in diesem Handbuch.

Nehmen Sie eine Rolle mit Web-Identität oder OpenID Connect an

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 APIReferenz unter [Aktionen](#).

Um Ihr SDK Tool für die Übernahme einer Rolle einzurichten, müssen Sie zunächst eine bestimmte Rolle erstellen oder identifizieren, die Sie übernehmen möchten. IAMRollen werden eindeutig durch eine Rolle identifiziert Amazon Resource Name ([ARN](#)). 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 im IAMBenutzerhandbuch unter [Methoden zur Übernahme einer Rolle](#).

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

Note

Es hat sich AWS bewährt, wann immer möglich regionale Endpunkte zu verwenden und Ihre [AWS-Region](#) zu konfigurieren.

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 zu erhalten `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 OIDC Discovery-Protokoll kompatibel sind, z. B. EntraID oder PingFederate.

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

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

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

Schritt 2: Konfigurieren Sie das Tool SDK oder

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

Wenn Sie dies in einem Profil angeben, führt das Tool SDK oder 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 Amazon Cognito zuzugreifen API, schließen Sie den Amazon Cognito Service Client 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.

AWS -Zugriffsschlüssel

Verwenden kurzfristiger Anmeldeinformationen

Wir empfehlen, Ihr SDK oder Tool so zu konfigurieren, dass es verwendet, um Optionen für die erweiterte Sitzungsdauer [IAM Identity Center-Authentifizierung für Ihr Tool SDK oder](#) zu verwenden.

Informationen zum direkten Einrichten der temporären Anmeldeinformationen des SDK oder Tools finden Sie unter [Authentifizieren Sie sich mit kurzfristigen Anmeldeinformationen](#).

Verwenden langfristiger 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 des Zugriffs über hinweg AWS-Konten

Als bewährte Sicherheitsmethode empfehlen wir die Verwendung von AWS Organizations mit IAM Identity Center, um den Zugriff auf all Ihre zu verwalten AWS-Konten. Weitere Informationen finden Sie unter [Bewährte Methoden für die Sicherheit in IAM](#) im IAM-Benutzerhandbuch.

Sie können Benutzer in IAM Identity Center erstellen, Microsoft Active Directory verwenden, einen SAML-2.0-Identitätsanbieter (IdP) verwenden oder Ihren IdP einzeln mit verbinden AWS-Konten. Mit einem dieser Ansätze können Sie Ihren Benutzern eine Single-Sign-On-Erfahrung bieten. Sie können auch die Multi-Faktor-Authentifizierung (MFA) erzwingen und temporäre Anmeldeinformationen für den AWS-Konto Zugriff verwenden. Dies unterscheidet sich von einem IAM-Benutzer, bei dem es sich um langfristige Anmeldeinformationen handelt, die freigegeben werden können und das Sicherheitsrisiko für Ihre AWS Ressourcen erhöhen können.

Erstellen von IAM-Benutzern nur für Sandbox-Umgebungen

Wenn Sie noch nicht mit vertraut sind AWS, können Sie einen IAM-Testbenutzer erstellen und ihn dann verwenden, um Tutorials auszuführen und zu erfahren, was zu bieten AWS ist. Es ist in

Ordnung, diese Art von Anmeldeinformationen beim Lernen zu verwenden, aber wir empfehlen, sie nicht außerhalb einer Sandbox-Umgebung zu verwenden.

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

- Erste Schritte mit Ihrem AWS SDK oder Tool und Erkunden AWS-Services in einer Sandbox-Umgebung.
- Ausführen geplanter Skripts, Aufträge und anderer automatisierter Prozesse, die im Rahmen Ihres Lernens keinen beaufsichtigten Anmeldeprozess unterstützen.

Wenn Sie IAM-Benutzer außerhalb dieser Anwendungsfälle verwenden, wechseln Sie AWS-Konten so schnell wie möglich zum IAM Identity Center oder verbinden Sie Ihren Identitätsanbieter mit . Weitere Informationen finden Sie unter [Identitätsverbund in AWS](#).

Sichere IAM-Benutzerzugriffsschlüssel

Sie sollten die Zugriffsschlüssel von IAM-Benutzern regelmäßig rotieren. Folgen Sie den Anweisungen unter [Rotieren von Zugriffsschlüsseln](#) im IAM-Benutzerhandbuch. Wenn Sie glauben, dass Sie versehentlich Ihre IAM-Benutzerzugriffsschlüssel freigegeben haben, rotieren Sie Ihre Zugriffsschlüssel.

IAM-Benutzerzugriffsschlüssel sollten in der AWS `credentials` freigegebenen Datei auf dem lokalen Computer gespeichert werden. Speichern Sie die IAM-Benutzerzugriffsschlüssel nicht in Ihrem Code. Schließen Sie keine Konfigurationsdateien ein, die Ihre IAM-Benutzerzugriffsschlüssel in einer Quellcodeverwaltungssoftware enthalten. Externe Tools wie die Open-Source-Projekt-[git-secrets](#) können Ihnen helfen, versehentlich vertrauliche Informationen in ein Git-Repository zu übertragen. Weitere Informationen finden Sie unter [IAM-Identitäten \(Benutzer, Gruppen und Rollen\)](#) im IAM-Benutzerhandbuch.

Informationen zum Einrichten eines IAM-Benutzers für die ersten Schritte finden Sie unter [Authentifizieren Sie sich mit langfristigen Anmeldeinformationen](#).

Authentifizieren Sie sich mit kurzfristigen Anmeldeinformationen

Wir empfehlen, Ihr SDK OR-Tool so zu konfigurieren, dass es [IAM Identity Center-Authentifizierung für Ihr Tool SDK oder](#) mit Optionen für die erweiterte Sitzungsdauer verwendet werden kann. 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 [IAMIdentity Center-Authentifizierung für Ihr Tool SDK oder](#) z.
- Oder verwenden Sie die [Anbieter von Prozessanmeldedaten](#), 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 Tool SDK oder 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 IAM Rollenmeldedaten 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`.

Wichtige Warnhinweise und Richtlinien für Anmeldeinformationen

Warnhinweise für Anmeldeinformationen

- **NOT** Verwenden Sie die Root-Anmeldeinformationen Ihres Kontos, um auf AWS Ressourcen zuzugreifen. Diese Anmeldeinformationen bieten uneingeschränkten Zugriff auf Konten und können nur schwer widerrufen werden.
- **NOT** Geben Sie wörtliche 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.
- **NOT** Fügen Sie Dateien mit Anmeldeinformationen in Ihren Projektbereich ein.
- Beachten Sie, dass alle in der gemeinsam genutzten `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](#) für Amazon Elastic Container Service (Amazon ECS) - Aufgaben.
- Verwenden Sie [IAM Rollen](#) für Anwendungen, die auf EC2 Amazon-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 AWS Erstbenutzer?](#) im AWS Account Management 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 im Abschnitt [Erstellen eines Administratorbenutzers](#) im IAMBenutzerhandbuch 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 geringsten Berechtigungen](#) im IAM-Benutzerhandbuch.

Schritt 1: Erstellen Sie Ihren IAM Benutzer

- Erstellen Sie Ihren IAM Benutzer, indem Sie den Anweisungen [zum Erstellen von IAM Benutzern \(Konsole\)](#) im IAMBenutzerhandbuch folgen. Gehen Sie beim Erstellen Ihres IAM Benutzers wie folgt vor:
 - Wir empfehlen Ihnen, Benutzerzugriff auf die bereitzustellen auszuwählen AWS Management Console. 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 „Erste Schritte“ SDK -Tutorials verwenden den Amazon S3 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 zur Festlegung von Berechtigungsgrenzen oder Tags ignorieren.

Schritt 2: Abrufen Ihrer Zugriffsschlüssel

1. Wählen Sie im Navigationsbereich der IAM Konsole Benutzer und dann den **User name** Benutzer aus, den Sie zuvor erstellt haben.

- Wählen Sie auf der Seite des Benutzers die Seite Sicherheitsanmeldeinformationen aus. Wählen Sie dann unter Zugriffsschlüssel die Option Zugriffsschlüssel erstellen aus.
- 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.
- Geben Sie für Zugriffsschlüssel erstellen – Schritt 2 ein optionales Tag ein und wählen Sie Weiter aus.
- Wählen Sie für Schritt 3 „Zugriffsschlüssel erstellen“ 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.

- Wählen Sie Done (Fertig).

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

- 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](#).
- 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
```

- 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 unter [AWS Zugriffstasten](#). Dies ist eine Möglichkeit, Ihnen den Einstieg zu erleichtern, aber wir empfehlen Ihnen, so bald wie möglich zu IAM Identity Center oder anderen temporären Anmeldeinformationen zu wechseln. Denken Sie nach der Umstellung auf die Verwendung langfristiger Anmeldeinformationen daran, diese Anmeldeinformationen aus der gemeinsam genutzten `credentials` Datei zu löschen.

IAMRollen für EC2 Amazon-Instances verwenden

Dieses Beispiel behandelt die Einrichtung einer AWS Identity and Access Management Rolle mit Amazon S3 S3-Zugriff zur Verwendung in Ihrer auf einer EC2 Amazon-Instance bereitgestellten Anwendung.

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 [IAMRollen für Amazon EC2](#) im EC2Amazon-Benutzerhandbuch.

Erstellen Sie eine IAM-Rolle

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

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

1. Melden Sie sich bei der an AWS Management Console 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 mithilfe Ihrer IAM Rolle eine EC2 Amazon-Instance zu erstellen und zu starten:

- Folgen Sie [Quickly launch an instance](#) im EC2Amazon-Benutzerhandbuch. Gehen Sie vor dem letzten Einreichungsschritt jedoch auch wie folgt vor:
 - Wählen Sie unter Erweiterte Details für IAMInstanzprofil die Rolle aus, die Sie im vorherigen Schritt erstellt haben.

Mit dieser IAM und der EC2 Einrichtung von Amazon 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 Management Console 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 IAMRolle ändern aus.
 - d. Trennen Sie die IAMRolle von der IAM Rolle, indem Sie „Keine IAM Rolle“ auswählen.
 - e. Wählen Sie IAMRolle aktualisieren aus.
 - f. Führen Sie die Anwendung erneut aus und vergewissern Sie sich, dass sie einen Autorisierungsfehler zurückgibt.

Referenz zu Einstellungen

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 SDKs standardisierte Funktionen und Tools](#)— Gemeinsame Funktionen, die für mehrere SDKs standardisiert sind.

Serviceclients erstellen

SDKs Verwenden Sie für den programmgesteuerten Zugriff AWS-Services jeweils eine Clientklasse/ ein Client-Objekt. 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 for PowerShell es sich um Parameter für einzelne Operationen, 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 instanziiieren, 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 for .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 SDK oder Tool, das Sie verwenden.

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.

Seiten mit Einstellungen

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.

- [ConfigListe der Dateieinstellungen](#)
- [CredentialsListe 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` Beispielseinstellung vermitteln 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 `false` gesetzt.
- Die einzigen gültigen Werte für diese boolesche Einstellung sind `true` und `false`

Am Ende jeder Feature-Seite befindet sich eine Tabelle zur Kompatibilität mit AWS SDKs.

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.

ConfigListe 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 eindeutige 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
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_request_compression	Komprimierung anfordern
duration_seconds	Nehmen Sie die Rolle des Anbieters von Anmeldeinformationen an

Einstellungsname	Details
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
metadata_service_num_attempts	EC2 Amazon-Instanz-Metadaten
metadata_service_timeout	EC2 Amazon-Instanz-Metadaten

Einstellungsname	Details
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_minimum_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
role_session_name	Nehmen Sie die Rolle des Anbieters von Anmeldeinformationen an
s3_disable_multiregion_access_points	Multiregionale Amazon-S3-Zugriffspunkte

Einstellungsname	Details
s3_use_arn_region	Amazon-S3-Zugriffspunkte
sdk_ua_app_id	Application ID
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
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

CredentialsListe 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 eindeutige 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
<code>aws_access_key_id</code>	AWS Zugriffstasten
<code>aws_secret_access_key</code>	AWS Zugriffstasten
<code>aws_session_token</code>	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 eindeutige 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
<code>AWS_ACCESS_KEY_ID</code>	AWS Zugriffstasten
<code>AWS_ACCOUNT_ID</code>	Kontobasierte Endpunkte
<code>AWS_ACCOUNT_ID_ENDPOINT_MODE</code>	Kontobasierte Endpunkte

Einstellungsname	Details	
AWS_CA_BUNDLE	Allgemeine Konfigurationseinstellungen	
AWS_CONFIG_FILE	Speicherort der geteilten <code>credentials</code> Dateien <code>config</code> und Dateien	
AWS_CONTAINER_AUTHORIZATION_TOKEN	Anbieter von Container-Anmeldeinformationen	
AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE	Anbieter von Container-Anmeldeinformationen	
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_REQUEST_COMPRESSION	Komprimierung anfordern	
AWS_EC2_METADATA_DISABLED	Anbieter von IMDS-Anmeldeinformationen	

Einstellungsname	Details
AWS_EC2_METADATA_SERVICE_ENDPOINT	Anbieter von IMDS-Anmeldeinformationen
AWS_EC2_METADATA_ENDPOINT_MODE	Anbieter von IMDS-Anmeldeinformationen
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_ENDPOINT_URLS	Servicespezifische Endpunkte
AWS_MAX_ATTEMPTS	Verhalten wiederholen

Einstellungsname	Details
AWS_METAD ATA_SERVI CE_NUM_AT TEMPTS	EC2 Amazon-Instanz-Metadaten
AWS_METAD ATA_SERVI CE_TIMEOUT	EC2 Amazon-Instanz-Metadaten
AWS_PROFILE	Geteilte credentials Dateien config und Dateien
AWS_REGION	AWS-Region
AWS_REQUE ST_CHECKS UM_CALCULATION	Schutz der Datenintegrität für Amazon S3
AWS_REQUE ST_MIN_CO MPRESSION _SIZE_BYTES	Komprimierung anfordern
AWS_RESPO NSE_CHECK SUM_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

Einstellungsname	Details
AWS_S3_DISTRIBUTABLE_MULTIREGION_ACCESS_POINTS	Multiregionale Amazon-S3-Zugriffspunkte
AWS_S3_US_EARN_REGION	Amazon-S3-Zugriffspunkte
AWS_SDK_USER_APP_ID	Application ID
AWS_SECRET_ACCESS_KEY	AWS Zugriffstasten
AWS_SESSION_TOKEN	AWS Zugriffstasten
AWS_SHARED_CREDENTIALS_FILE	Speicherort der geteilten credentials Dateien config und Dateien
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 for Java und die AWS SDK for 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
<code>aws.accessKeyId</code>	AWS Zugriffstasten
<code>aws.accountId</code>	Kontobasierte Endpunkte
<code>aws.accountIdEndpointMode</code>	Kontobasierte Endpunkte
<code>aws.configFile</code>	Speicherort der geteilten Dateien und Dateien <code>configcredentials</code>
<code>aws.defaultsMode</code>	Standardeinstellungen für die intelligente Konfiguration
<code>aws.disableEc2MetadataV1</code>	Anbieter von IMDS-Anmeldeinformationen
<code>aws.disableRequestCompression</code>	Komprimierung anfordern
<code>aws.ec2MetadataServiceEndpoint</code>	Anbieter von IMDS-Anmeldeinformationen
<code>aws.ec2MetadataServiceEndpointMode</code>	Anbieter von IMDS-Anmeldeinformationen

Einstellungsname	Details
<code>aws.endpointDiscoveryEnabled</code>	Erkennung von Endpunkten
<code>aws.endpointUrl</code>	Servicespezifische Endpunkte
<code>aws.endpointUrl<ServiceName></code>	Servicespezifische Endpunkte
<code>aws.ignoreConfiguredEndpointUrls</code>	Servicespezifische Endpunkte
<code>aws.maxAttempts</code>	Verhalten wiederholen
<code>aws.profile</code>	Geteilte Dateien config und Dateien credentials
<code>aws.region</code>	AWS-Region
<code>aws.requestChecksumCalculation</code>	Schutz der Datenintegrität für Amazon S3
<code>aws.requestMinCompressionSizeBytes</code>	Komprimierung anfordern
<code>aws.responseChecksumValidation</code>	Schutz der Datenintegrität für Amazon S3
<code>aws.retryMode</code>	Verhalten wiederholen

Einstellungsname	Details
<code>aws.roleArn</code>	Nehmen Sie die Rolle des Anbieters von Anmeldeinformationen an
<code>aws.roleSessionName</code>	Nehmen Sie die Rolle des Anbieters von Anmeldeinformationen an
<code>aws.s3DisableMultiRegionAccessPoints</code>	Multiregionale Amazon-S3-Zugriffspunkte
<code>aws.s3UseArnRegion</code>	Amazon-S3-Zugriffspunkte
<code>aws.secretAccessKey</code>	AWS Zugriffstasten
<code>aws.sessionToken</code>	AWS Zugriffstasten
<code>aws.sharedCredentialsFile</code>	Speicherort der geteilten credentials Dateien config und Dateien
<code>aws.useDualstackEndpoint</code>	Dual-Stack- und FIPS-Endpunkte
<code>aws.useFipsEndpoint</code>	Dual-Stack- und FIPS-Endpunkte
<code>aws.userAgentAppId</code>	Application ID
<code>aws.webIdentityTokenFile</code>	Nehmen Sie die Rolle des Anbieters von Anmeldeinformationen an

AWS SDKs und Tools standardisierte Anbieter von Anmeldeinformationen

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

Important

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

Themen

- [Verstehen Sie die Kette der Anbieter von Anmeldeinformationen](#)
- [SDK-spezifische und toolspezifische Anbieterketten für Anmeldeinformationen](#)
- [AWS Zugriffstasten](#)
- [Übernehmen Sie die Rolle Credential Provider](#)
- [Anbieter von Container-Anmeldeinformationen](#)
- [IAM Identity Center-Anmeldeinformationsanbieter](#)
- [IMDS Anbieter von Anmeldeinformationen](#)
- [Anbieter von Prozessanmeldedaten](#)

Verstehen Sie die Kette der Anbieter von Anmeldeinformationen

Alle SDKs haben eine Reihe von Stellen (oder Quellen), an denen sie nach gültigen Anmeldeinformationen suchen, mit denen sie eine Anfrage an AWS-Service einen stellen können. 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 diese AWS SDKs immer, Anmeldeinformationen automatisch zu erneuern, wenn sie ablaufen. Die integrierte Anmeldeinformationsanbieterkette bietet Ihrer Anwendung die Möglichkeit, Ihre Anmeldeinformationen unabhängig davon zu aktualisieren, welchen Anbieter Sie in der Kette verwenden. Dazu ist kein zusätzlicher Code erforderlich. SDK

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

Anbieter von Anmeldeinformationen	Beschreibung
AWS Zugriffstasten	AWS Zugriffstasten für einen IAM Benutzer (wie <code>AWS_ACCESS_KEY_ID</code> , und <code>AWS_SECRET_ACCESS_KEY</code>).
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 Web-Token (JWT) von () verwenden. AWS Security Token Service AWS STS
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 von Container-Anmeldeinformationen ruft Anmeldeinformationen für die containerisierte Anwendung des Kunden ab.
Anbieter von Prozessanmeldedaten	Benutzerdefinierter Anbieter für Anmeldeinformationen. Rufen Sie Ihre Anmeldeinformationen aus einer externen Quelle oder einem externen Prozess ab, einschließlich IAM Roles Anywhere.
IMDS Anbieter von Anmeldeinformationen	Anmeldeinformationen für das Amazon Elastic Compute Cloud (Amazon EC2) -Instanzprofil. Ordnen Sie jeder Ihrer EC2 Instances eine IAM Rolle zu. Temporäre

Anbieter von Anmeldeinformationen	Beschreibung
	Anmeldeinformationen für diese Rolle werden dem Code zur Verfügung gestellt, der in der Instanz ausgeführt wird. Die Anmeldeinformationen werden über den EC2 Amazon-Metadatenservice 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 die [Geteilte credentials Dateien config und Dateien](#). Weitere Informationen finden Sie unter [Vorrang der Einstellungen](#).

SDK-spezifische und toolspezifische Anbieterketten für Anmeldeinformationen

Um direkt zu den Details der Kette der Anbieter SDK von Anmeldedaten oder zu den spezifischen Zugangsdatenanbietern Ihres Tools zu gelangen, wählen Sie Ihr Tool SDK oder aus den folgenden Optionen aus:

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

AWS Zugriffstasten

Warning

Verwenden Sie zur Vermeidung von Sicherheitsrisiken keine IAM Benutzer zur Authentifizierung, 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. The verwendet diese AWS Anmeldeinformationen AWS SDK 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 -spezifischen Einstellungen der gemeinsam genutzten Datei getrennt ist. `config`

Weitere Informationen zu AWS Anmeldeinformationen und zur Verwendung von Zugriffsschlüsseln finden Sie unter [AWS Sicherheitsanmeldeinformationen](#) und [Verwaltung von Zugriffsschlüsseln für IAM Benutzer](#) im IAMBenutzerhandbuch.

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 unter IAM](#).

Anweisungen zum Abrufen dieser Werte finden Sie unter [Authentifizieren Sie sich mit kurzfristigen Anmeldeinformationen](#).

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...zrkuWJ0gQs8IZZaIv2BXIa2R40Lgk
```

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

```
export AWS_ACCESS_KEY_ID=AKIAIOSFODNN7EXAMPLE
export AWS_SECRET_ACCESS_KEY=wJa1rXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY
export
  AWS_SESSION_TOKEN=AQoEXAMPLEH4aoAH0gNCAPy...truncated...zrkuWJ0gQs8IZZaIv2BXIa2R40lgk
```

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

```
setx AWS_ACCESS_KEY_ID AKIAIOSFODNN7EXAMPLE
setx AWS_SECRET_ACCESS_KEY wJa1rXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY
setx
  AWS_SESSION_TOKEN AQoEXAMPLEH4aoAH0gNCAPy...truncated...zrkuWJ0gQs8IZZaIv2BXIa2R40lgk
```

Kompatibilität mit AWS SDKs

Im Folgenden werden die in diesem Thema beschriebenen Funktionen und Einstellungen SDKs unterstützt. Alle teilweisen Ausnahmen werden vermerkt. Alle Einstellungen für JVM Systemeigenschaften werden AWS SDK for Kotlin nur von AWS SDK for Java und vom unterstützt.

SDK	U zt	Hinweise oder weitere Informationen
AWS CLI v2	Ja	
SDK für C++	Ja	gemeinsam genutzte config Datei wird nicht unterstützt.
SDK für Go V2 (1.x)	Ja	
SDK für 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 für Java 2.x	Ja	
SDK für Java 1.x	Ja	
SDK für 3.x JavaScript	Ja	
SDK für 2.x JavaScript	Ja	

SDK	U zt	Hinweise oder weitere Informationen
SDK für Kotlin	Ja	
SDK für .NET 3.x	Ja	Umgebungsvariablen werden nicht unterstützt.
SDK für PHP 3.x	Ja	
SDK für Python (Boto3)	Ja	
SDK für Ruby 3.x	Ja	
SDK für Rust	Ja	
SDK für Swift	Ja	
Tools für PowerShell	Ja	Umgebungsvariablen werden nicht unterstützt.

Übernehmen Sie die Rolle Credential Provider

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 Tool für die Übernahme einer Rolle einzurichten, müssen Sie zunächst eine bestimmte Rolle erstellen oder identifizieren, die Sie übernehmen möchten. IAM Rollen werden eindeutig durch eine Rolle identifiziert Amazon Resource Name ([ARN](#)). Rollen bauen Vertrauensbeziehungen zu einer anderen Entität auf. Bei der vertrauenswürdigen Entität, die die Rolle verwendet AWS-Service, kann es sich um ein AWS-Konto, ein anderes, einen Web-Identitätsanbieter oder einen OIDC SAML Verbund handeln.

Nachdem die IAM Rolle identifiziert wurde und Sie aufgrund dieser Rolle vertrauenswürdig sind, können Sie Ihr SDK Tool so konfigurieren, dass es die von der Rolle gewährten Berechtigungen verwendet. Verwenden Sie dazu die folgenden Einstellungen.

Anleitungen zu den ersten Schritten mit diesen Einstellungen finden Sie [Nehmen Sie eine Rolle mit AWS Anmeldeinformationen an](#) 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 EC2 Amazon-Instances oder Amazon Elastic Container Service-Containern verwendet, um anzugeben, wo das Tool SDK oder 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 Tool SDK oder Quellanmeldedaten aus den Umgebungsvariablen [AWS_ACCESS_KEY_ID](#) und [AWS_SECRET_ACCESS_KEY](#) abrufen soll.
- `Ec2 InstanceMetadata` — Gibt an, dass das Tool SDK oder die dem [EC2 Instanzprofil zugeordnete IAM Rolle zum Abrufen der](#) Quellanmeldedaten verwenden soll.
- `EcsContainer` — Gibt an, dass das Tool SDK oder die dem [ECS Container zugeordnete IAM Rolle zum Abrufen der](#) Quellanmeldedaten verwenden soll.

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 bezogen werden sollten EC2:

```
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 `requiresExternalId`. 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 eine Rolle übernommen wird, bei der die Vertrauensrichtlinie für diese Rolle eine Bedingung beinhaltet, die eine MFA Authentifizierung erfordert. Weitere Informationen MFA dazu finden Sie unter [AWS Multi-Faktor-Authentifizierung 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 von 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 `MyMFADevice`, dass ein virtuelles MFA Gerät namens, 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 ARN einer IAM Rolle entsprechen und wie folgt formatiert sein: `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 an ein EC2 Amazon-Instance-Profil angehängt sind, oder eine ECS Amazon-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 [CloudTrail userIdentity 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 der AWS CLI oder die `AssumeRole` Operation (oder Operationen wie die `AssumeRoleWithWebIdentity` Operation) in Ihrem Namen AWS API 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 in der `role_arn` Einstellung im ursprünglichen Profil angegebene Rolle anzunehmen. Informationen zur Verwendung von Profilen in geteilten `credentials` Dateien AWS `config` und Dateien finden Sie unter [Geteilte 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 der SDK auf ein Profil mit Anmeldeinformationen trifft. Die Rollenverkettung begrenzt Ihre Sitzung AWS CLI oder Ihre AWS API Rollensitzung auf maximal eine Stunde und kann nicht verlängert werden. Weitere Informationen finden Sie im IAMBenutzerhandbuch unter [Begriffe und Konzepte für Rollen](#).

Standardwert: Keiner.

Gültige Werte: Eine Textzeichenfolge, die aus dem Namen eines in den `credentials` Dateien `config` und definierten Profils besteht. 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 Tool SDK oder an, automatisch nach den Anmeldeinformationen für das verknüpfte B Profil zu suchen. In diesem Fall verwendet das B Profil das Credential Helper-Tool von, [IAM Roles Anywhere](#) um die Anmeldeinformationen für abzurufen. AWS SDK 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 ausgeführt wird, A 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 EC2 Amazon-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 Entwicklertool SDK oder 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.

Kompatibilität mit AWS SDKs

Im Folgenden werden die in diesem Thema beschriebenen Funktionen und Einstellungen SDKs unterstützt. Alle teilweisen Ausnahmen werden vermerkt. Alle Einstellungen für JVM Systemeigenschaften werden AWS SDK for Kotlin nur von AWS SDK for Java und vom unterstützt.

SDK	U	Hinweise oder weitere Informationen
AWS CLI v2	Ja	
SDK für C++	Teilwe	<code>credential_source</code> wird nicht unterstützt. <code>duration_seconds</code> nicht unterstützt. <code>mfa_serial</code> nicht unterstützt.
SDK für Go V2 (1.x)	Ja	
SDK für 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 für Java 2.x	Teilwe	<code>mfa_serial</code> wird nicht unterstützt. <code>duration_seconds</code> nicht unterstützt.
SDK für Java 1.x	Teilwe	<code>credential_source</code> wird nicht unterstützt. <code>mfa_serial</code> nicht unterstützt. JVM Systemeigenschaften werden nicht unterstützt.
SDK für JavaScript 3.x	Ja	
SDK für 2.x JavaScript	Teilwe	<code>credential_source</code> nicht unterstützt.
SDK für Kotlin	Ja	

SDK	U zt	Hinweise oder weitere Informationen
SDK für .NET 3.x	Ja	
SDK für 3.x PHP	Ja	
SDK für Python (Boto3)	Ja	
SDK für Ruby 3.x	Ja	
SDK für Rust	Ja	
SDK für Swift	Ja	
Tools für PowerShell	Ja	

Anbieter von Container-Anmeldeinformationen

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

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

Wenn Sie Amazon verwenden, empfehlen wir Ihnen EKS, 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 Ihr 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 EKS Amazon-Benutzerhandbuch. Nach der Konfiguration EKS legt

Amazon die Umgebungsvariablen `AWS_CONTAINER_CREDENTIALS_FULL_URI` und die `AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE` Umgebungsvariablen fest, die die SDKs Tools zum Abrufen von Anmeldeinformationen verwenden. Informationen zur Einrichtung finden Sie unter [Einrichtung des Amazon EKS Pod Identity Agent](#) im EKSAmerican-Benutzerhandbuch oder [Amazon EKS Pod Identity vereinfacht IAM Berechtigungen für Anwendungen auf EKS Amazon-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 SDK, der bei der Anforderung von Anmeldeinformationen verwendet werden soll. Dies umfasst sowohl das Schema als auch den Host.

Standardwert: Keiner.

Gültige Werte: GültigURI.

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
```

or

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

AWS_CONTAINER_CREDENTIALS_RELATIVE_URI- Umgebungsvariable

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

Standardwert: Keiner.

Gültige Werte: Gültiger VerwandterURI.

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, SDK wird der Authorization-Header der HTTP Anfrage mit dem Wert der Umgebungsvariablen gesetzt.

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
```

Kompatibilität mit AWS SDKs

Im Folgenden werden die in diesem Thema beschriebenen Funktionen und Einstellungen SDKs unterstützt. Alle teilweisen Ausnahmen werden vermerkt. Alle Einstellungen für JVM Systemeigenschaften werden AWS SDK for Kotlin nur von AWS SDK for Java und vom unterstützt.

SDK	U zt	Hinweise oder weitere Informationen
AWS CLI v2	Ja	
SDK für C++	Ja	
SDK für Go V2 (1.x)	Ja	
SDK für Go 1.x (V1)	Ja	
SDK für Java 2.x	Ja	AWS_CONTAINER_CREDENTIALS_FULL_URI und AWS_CONTAINER_AUTHORIZATION_TOKEN werden auch für Lambda SnapStart für Java verwendet.
SDK für Java 1.x	Ja	AWS_CONTAINER_CREDENTIALS_FULL_URI und AWS_CONTAINER_AUTHORIZATION_TOKEN werden auch für Lambda SnapStart für Java verwendet.
SDK für 3.x JavaScript	Ja	
SDK für 2.x JavaScript	Ja	
SDK für Kotlin	Ja	
SDK für .NET 3.x	Ja	
SDK für 3.x PHP	Ja	
SDK für Python (Boto3)	Ja	
SDK für Ruby 3.x	Ja	
SDK für Rust	Ja	
SDK für Swift	Ja	
Tools für PowerShell	Ja	

IAM Identity Center-Anmeldeinformationsanbieter

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 Credential Provider bezeichnet.

Nachdem Sie IAM Identity Center aktiviert haben, definieren Sie ein Profil für die zugehörigen Einstellungen in Ihrer gemeinsam genutzten AWS `config` Datei. Dieses Profil wird verwendet, um eine Verbindung zum IAM Identity Center-Zugriffportal herzustellen. Wenn sich ein Benutzer erfolgreich bei IAM Identity Center authentifiziert, gibt das Portal kurzfristige Anmeldeinformationen für die diesem Benutzer zugeordnete IAM Rolle zurück. Informationen darüber, wie der temporäre Anmeldeinformationen aus der Konfiguration SDK erhält und sie für AWS-Service Anfragen verwendet, finden Sie unter [Verstehen Sie die IAM Identity Center-Authentifizierung](#).

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

- (Empfohlene) 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 API Portal-Referenz](#)

Ausführliche Informationen darüber, wie die Tools SDKs und die Anmeldeinformationen mithilfe dieser Konfiguration verwenden und aktualisieren, finden Sie unter [Verstehen Sie die IAM Identity Center-Authentifizierung](#).

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 Einrichtung [IAM Identity Center-Authentifizierung für Ihr Tool SDK oder](#).

SSOKonfiguration des Token-Anbieters

Wenn Sie die SSO Token-Provider-Konfiguration verwenden, aktualisiert Ihr AWS SDK 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 unter [Format der Konfigurationsdatei](#).

Im folgenden Beispiel für eine gemeinsam genutzte `config` Datei wird das Tool SDK oder 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
```

Die vorherigen Beispiele zeigen, dass Sie einen `sso-session` Abschnitt definieren und ihn einem Profil zuordnen. Normalerweise `sso_account_id` und `sso_role_name` muss in dem `profile` Abschnitt festgelegt werden, damit sie AWS Anmeldeinformationen anfordern SDK können. `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. Die Bearer-Authentifizierung ist ein HTTP Authentifizierungsschema, das Sicherheitstoken verwendet, die als Bearer-Token bezeichnet werden. 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` Geltungsbereich ist ein Mechanismus in OAuth 2.0 um den Zugriff einer Anwendung auf das Konto eines Benutzers einzuschränken. Im vorherigen Beispiel wurde festgelegt `sso_registration_scopes`, dass der erforderliche Zugriff für die Auflistung von Konten und Rollen bereitgestellt werden soll.

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, [SSOKonfiguration des Token-Anbieters](#) stattdessen das zu verwenden.

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.

IAMEinstellungen des Identity Center-Anmeldeinformationsanbieters

Konfigurieren Sie diese Funktionalität wie folgt:

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

Die URL, die auf den IAM Identity Center-Aussteller URL oder das Zugriffsportal URL 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 dem AWS Zugangsportal URL.

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

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 [IAMIdentity 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 [IAMIdentity Center-Konsole](#) und öffnen Sie die AWS-KontenSeite. Sie können die Liste der verfügbaren Konten mit dieser [ListAccounts](#)APIMethode auch in der AWS IAM Identity Center APIPortalreferenz einsehen. Sie können beispielsweise die AWS CLI Methode [list-accounts](#) aufrufen.

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

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

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

Um die Liste der verfügbaren Berechtigungssätze pro zu sehen AWS-Konto, gehen Sie zur [IAMIdentity Center-Konsole](#) und öffnen Sie die AWS-KontenSeite. Wählen Sie den richtigen Namen für den Berechtigungssatz, der in der AWS-Konten Tabelle aufgeführt ist. Sie können die Liste der verfügbaren Berechtigungssätze, die [ListAccountRoles](#)APIdiese Methode verwenden, auch in der AWS IAM Identity Center APIPortalreferenz einsehen. Sie können die AWS CLI Methode beispielsweise aufrufen [list-account-roles](#).

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

Eine durch Kommas getrennte Liste gültiger Bereichszeichenfolgen, für die autorisiert werden sollen. `sso-session` Eine Anwendung kann einen oder mehrere Bereiche anfordern, und das für die Anwendung ausgegebene Zugriffstoken ist auf die gewährten Bereiche beschränkt. Ein Mindestumfang 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 unter [Zugriffsbereiche](#) im AWS IAM Identity Center Benutzerhandbuch.

Diese Bereiche definieren die für die Autorisierung des registrierten OIDC Clients angeforderten Berechtigungen und die vom Client abgerufenen Zugriffstoken. Bereiche autorisieren den Zugriff auf autorisierte Endpunkte mit IAM Identity Center-Inhabertoken.

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

Kompatibilität mit AWS SDKs

Im Folgenden werden die in diesem Thema beschriebenen Funktionen und Einstellungen SDKs unterstützt. Alle teilweisen Ausnahmen werden vermerkt. Alle Einstellungen für JVM Systemeigenschaften werden AWS SDK for Kotlin nur von AWS SDK for Java und vom unterstützt.

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

SDK	U zt	Hinweise oder weitere Informationen
SDK für Java 1.x	Nein	
SDK für 3.x JavaScript	Ja	
SDK für 2.x JavaScript	Ja	
SDK für Kotlin	Ja	
SDK für .NET 3.x	Ja	
SDK für 3.x PHP	Ja	
SDK für Python (Boto3)	Ja	
SDK für Ruby 3.x	Ja	
SDK für Rust	Teilwe	Nur ältere, nicht aktualisierbare Konfiguration.
SDK für Swift	Ja	
Tools für PowerShell	Ja	

IMDS Anbieter von Anmeldeinformationen

Der Instanz-Metadatendienst (IMDS) stellt Daten über Ihre Instance bereit, mit denen Sie die laufende Instance konfigurieren oder verwalten können. Weitere Informationen zu den verfügbaren Daten finden Sie unter [Arbeiten mit Instance-Metadaten](#) im EC2 Amazon-Benutzerhandbuch. Amazon EC2 stellt einen lokalen Endpunkt bereit, 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. Sie SDKs können diesen Endpunkt verwenden, um Anmeldeinformationen als Teil ihrer [standardmäßigen Anbieterkette für Anmeldeinformationen aufzulösen](#). Instance Metadata Service Version 2 (IMDSv2), eine sicherere Version IMDS, die ein Sitzungstoken verwendet, wird standardmäßig verwendet. Wenn dies aufgrund eines Zustands fehlschlägt, der nicht erneut versucht werden kann (HTTP Fehlercodes 403, 404, 405), wird dies als Fallback verwendet.

Konfigurieren Sie diese Funktionalität wie folgt:

AWS_EC2_METADATA_DISABLED- Umgebungsvariable

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

Standardwert: `false`.

Zulässige Werte:

- **true**— Nicht IMDS zum Abrufen von Anmeldeinformationen verwenden.
- **false**— Wird verwendet 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

Ob Instance Metadata Service Version 1 (IMDSv1) als Fallback verwendet werden soll oder nicht, falls ein Fehler auftritt. IMDSv2

Note

New unterstützt diese Einstellung SDKs nicht IMDSv1 und unterstützt sie daher auch nicht. Einzelheiten finden Sie in der Tabelle [Kompatibilität mit AWS SDKs](#).

Standardwert: `false`.

Zulässige Werte:

- **true**— Nicht IMDSv1 als Fallback verwenden.
- **false**— IMDSv1 Als Fallback verwenden.

ec2_metadata_service_endpoint- Einstellung für gemeinsam genutzte AWS **config** Dateien, **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 EC2 Amazon-Instance-Metadaten suchen.

Standardwert: Wenn `ec2_metadata_service_endpoint_mode` gleich `IPv4`, dann ist der Standardendpunkt. `http://169.254.169.254` Wenn

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

Gültige Werte: GültigURI.

`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. [Verstehen Sie die Kette der Anbieter von Anmeldeinformationen](#) 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, geben Sie alternativ die Verwendung des IMDS Anmeldeinformationsanbieters im Code an. Sie können die Quellen für Anmeldeinformationen direkt angeben, wenn Sie Dienstclients erstellen.

Sicherheit für Anmeldeinformationen IMDS

Wenn der nicht mit gültigen Anmeldeinformationen konfiguriert AWS SDK ist, versucht er standardmäßig, den SDK 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 `true` gesetzt wird. 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 SDKClients, die mit gültigen Anmeldeinformationen konfiguriert sind, werden diese niemals verwenden, IMDS um Anmeldeinformationen abzurufen, unabhängig von diesen Einstellungen.

Verwendung von EC2 IMDS Amazon-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.

Kompatibilität mit AWS SDKs

Im Folgenden werden die in diesem Thema beschriebenen Funktionen und Einstellungen SDKs unterstützt. Alle teilweisen Ausnahmen werden vermerkt. Alle Einstellungen für JVM Systemeigenschaften werden AWS SDK for Kotlin nur von AWS SDK for Java und vom unterstützt.

SDK	Unterstützt	Hinweise oder weitere Informationen
AWS CLI v2	Ja	
SDK für C++	Ja	
SDK für Go V2 (1.x)	Ja	
SDK für 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 für Java 2.x	Ja	
SDK für Java 1.x	Teilwe	JVM Systemeigenschaften: Wird <code>com.amazonaws.sdk.disableEc2MetadataV1</code> anstelle von <code>aws.disableEc2MetadataV1</code> verwendet <code>aws.ec2MetadataServiceEndpoint</code> und wird <code>aws.ec2MetadataServiceEndpointMode</code> nicht unterstützt.
SDK für JavaScript 3.x	Ja	
SDK für 2.x JavaScript	Ja	
SDK für Kotlin	Ja	Verwendet kein IMDSv1 Fallback.
SDK für .NET 3.x	Ja	
SDK für 3.x PHP	Ja	
SDK für Python (Boto3)	Ja	
SDK für Ruby 3.x	Ja	

SDK	U zt	Hinweise oder weitere Informationen
SDK für Rust	Ja	Verwendet kein IMDSv1 Fallback.
SDK für Swift	Ja	
Tools für PowerShell	Ja	Sie können IMDSv1 Fallback explizit im Code deaktivieren, indem Sie <code>[Amazon.Util.EC2InstanceMetadata]::EC2MetadataV1Disabled = \$true</code> .

Anbieter von Prozessanmeldedaten

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.

IAM Roles Anywhere verwendet es beispielsweise, `credential_process` um temporäre Anmeldeinformationen für Ihre Anwendung abzurufen. Informationen zur Konfiguration `credential_process` für diese Verwendung finden Sie unter [IAM Roles Anywhere](#).

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 ausführen AWS. Wenn Sie auf einem bauen AWS Rechenressource, 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. Vergewissern Sie sich, dass Ihr benutzerdefiniertes Anmeldeinformationstool keine geheimen Informationen in das System schreibt `StdErr`, da das SDKs und AWS CLI kann solche Informationen erfassen und protokollieren, wodurch sie möglicherweise unbefugten Benutzern zugänglich gemacht werden.

Konfigurieren Sie diese Funktionalität wie folgt:

credential_process- geteilt AWS **config**Dateieinstellung

Gibt einen externen Befehl an, den das Tool SDK oder in Ihrem Namen ausführt, um die zu verwendenden Anmeldeinformationen zu generieren oder abzurufen. Die Einstellung gibt den Namen eines Programms oder Befehls an, das aufgerufen SDK wird. Wenn der den Prozess SDK aufruft, wartet er darauf, dass der Prozess Daten in den Prozess schreibt. JSON stdout Der benutzerdefinierte Anbieter muss Informationen in einem bestimmten Format zurückgeben. Diese Informationen enthalten die Anmeldeinformationen, mit denen das SDK OR-Tool Sie authentifizieren kann.

Note

Der Anbieter von Prozessanmeldedaten ist Teil von. [Verstehen Sie die Kette der Anbieter von Anmeldeinformationen](#) Der Anbieter für Prozessanmeldedaten wird jedoch erst nach mehreren anderen Anbietern aus dieser Serie geprüft. 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.

Geben Sie den Pfad zum Programm mit den Anmeldeinformationen an

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 diesen 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 die Einstellung von credentials al_process in der gemeinsam genutzten config Datei unter 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 generierenSTDOUT, 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"
```

}

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 RFC3339 formatierter Zeitstempel. Wenn der `Expiration` Schlüssel nicht in der Ausgabe des Tools enthalten ist, SDK wird davon ausgegangen, 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

Der SDK speichert die Anmeldeinformationen für externe Prozesse nicht im Cache, so wie er es bei der Übernahme von Rollenmeldedaten tut. Wenn Caching erforderlich ist, müssen Sie dies im externen Prozess implementieren.

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

Kompatibilität mit AWS SDKs

Im Folgenden werden die in diesem Thema beschriebenen Funktionen und Einstellungen SDKs unterstützt. Alle teilweisen Ausnahmen werden vermerkt. Alle Einstellungen für JVM Systemeigenschaften werden unterstützt von AWS SDK for Java und die AWS SDK for Kotlin nur.

SDK	U zt	Notizen oder weitere Informationen
AWS CLI v2	Ja	
SDK für C++	Ja	

SDK	U zt	Notizen oder weitere Informationen
SDK für Go V2 (1.x)	Ja	
SDK für Go 1.x (V1)	Ja	Um die Einstellungen für gemeinsam genutzte <code>config</code> Dateien zu verwenden, müssen Sie das Laden aus der Konfigurationsdatei aktivieren. Weitere Informationen finden Sie unter Sessions .
SDK für Java 2.x	Ja	
SDK für Java 1.x	Ja	
SDK für 3.x JavaScript	Ja	
SDK für 2.x JavaScript	Ja	
SDK für Kotlin	Ja	
SDK für .NET 3.x	Ja	
SDK für 3.x PHP	Ja	
SDK für Python (Boto3)	Ja	
SDK für Ruby 3.x	Ja	
SDK für Rust	Ja	
SDK für Swift	Ja	
Tools für PowerShell	Ja	

AWS SDKs standardisierte Funktionen und Tools

Viele Funktionen wurden auf einheitliche Standardwerte standardisiert und funktionieren bei 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

- [Kontobasierte Endpunkte](#)
- [Application ID](#)
- [EC2Amazon-Instanz-Metadaten](#)
- [Amazon S3 Access Points](#)
- [Multiregionale Amazon-S3-Zugriffspunkte](#)
- [AWS-Region](#)
- [AWS STS Regionale Endpunkte](#)
- [Schutz der Datenintegrität für Amazon S3](#)
- [Dual-Stack und Endpunkte FIPS](#)
- [Endpunkterkennung](#)
- [Allgemeine Konfigurationseinstellungen](#)
- [IMDSKlient](#)
- [Wiederholungsverhalten](#)
- [Komprimierung anfordern](#)
- [Servicespezifische Endpunkte](#)
- [Standardeinstellungen für intelligente Konfigurationen](#)

Kontobasierte Endpunkte

Kontobasierte Endpunkte sorgen für hohe Leistung und Skalierbarkeit, indem sie mithilfe Ihrer AWS-Konto ID die Weiterleitung von AWS-Service-Anfragen für Dienste, die diese Funktion unterstützen, optimieren. Wenn Sie einen AWS SDK-Anmeldeinformationsanbieter und einen Dienst verwenden, der kontobasierte Endpunkte unterstützt, erstellt und verwendet dieser SDK automatisch einen kontobasierten Endpunkt anstelle eines regionalen Endpunkts. Kontobasierte Endpunkte haben die Form von `https://<account-id>.ddb.<region>.amazonaws.com`, wo `<account-id>` wird durch Ihre ID ersetzt und wird durch Ihre AWS-Konto `<region>` AWS-Region

Standardmäßig wird die Konto-ID bei der Bearbeitung der Anfrage erfasst und zur Erstellung eines Endpunkts verwendet. Die Auflösung der Anmeldeinformationen erfolgt auch, wenn die Anfrage verarbeitet wird, und kann die Methode der Endpunktauflösung ändern. Je nachdem, welchen Anmeldeinformationsanbieter Sie verwenden, kann die Konto-ID aus unterschiedlichen Quellen stammen.

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

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

account_id_endpoint_mode- Einstellung für gemeinsam genutzte Dateien AWS **config**, **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, wird SDK ein Fehler ausgegeben.

Kompatibilität mit AWS SDKs

Im Folgenden werden die in diesem Thema beschriebenen Funktionen und Einstellungen SDKs unterstützt. Alle teilweisen Ausnahmen werden vermerkt. Alle Einstellungen für JVM Systemeigenschaften werden AWS SDK for Kotlin nur von AWS SDK for Java und vom unterstützt.

SDK	Unterstützt	In SDK Version veröffentlicht	Hinweise oder weitere Informationen
AWS CLI v2	Nein		
SDK für C++	Nein		
SDK für Go V2 (1.x)	Ja	v1.35.0	
SDK für Go 1.x (V1)	Nein		
SDK für Java 2.x	Ja	v2.28.4	
SDK für 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 für .NET 3.x	Nein		
SDK für 3.x PHP	Ja	v3.318.0	
SDK für Python (Boto3)	Nein		
SDK für Ruby 3.x	Ja	v1.123.0	
SDK für Rust	Nein		
SDK für Swift	Nein		
Tools für PowerShell	Nein		

Application ID

Eine einzige AWS-Konto kann von mehreren Kundenanwendungen verwendet werden, um Anrufe zu tätigen AWS-Services. Mithilfe der Anwendungs-ID können Kunden anhand eines ermitteln, welche Quellanwendung eine Reihe von Aufrufen getätigt hat AWS-Konto. AWS SDKs und Dienste verwenden oder interpretieren diesen Wert nur, um ihn in der Kundenkommunikation wieder zum Vorschein zu bringen. Dieser Wert kann beispielsweise in operativen E-Mails oder in der 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- geteilt AWS **config**Dateieinstellung, **AWS_SDK_UA_APP_ID**- Umgebungsvariable, **aws.userAgentAppId**- JVM Systemeigenschaft: Nur Java/Kotlin

Diese Einstellung ist eine eindeutige Zeichenfolge, die Sie Ihrer Anwendung zuweisen, um zu identifizieren, welche Ihrer Anwendungen in einer bestimmten Anwendung enthalten ist AWS-Konto ruft an 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.

Kompatibilität mit AWS SDKs

Im Folgenden werden die in diesem Thema beschriebenen Funktionen und Einstellungen SDKs unterstützt. Alle teilweisen Ausnahmen werden vermerkt. Alle Einstellungen für JVM Systemeigenschaften werden unterstützt von AWS SDK for Java und die AWS SDK for Kotlin nur.

SDK	U zt	Hinweise oder weitere Informationen
AWS CLI v2	Ja	
SDK für C++	Ja	gemeinsam genutzte config Datei wird nicht unterstützt.
SDK für Go V2 (1.x)	Ja	
SDK für Go 1.x (V1)	Nein	
SDK für Java 2.x	Teilwe	configDie Einstellung für gemeinsam genutzte Dateien wird nicht unterstützt; die Umgebungsvariable wird nicht unterstützt.
SDK für Java 1.x	Nein	
SDK für 3.x JavaScript	Ja	
SDK für 2.x JavaScript	Nein	
SDK für Kotlin	Ja	
SDK für .NET 3.x	Ja	Umgebungsvariablen werden nicht unterstützt.
SDK für PHP 3.x	Ja	
SDK für Python (Boto3)	Ja	
SDK für Ruby 3.x	Ja	
SDK für Rust	Ja	

SDK	U zt	Hinweise oder weitere Informationen
SDK für Swift	Nein	
Tools für PowerShell	Nein	

EC2 Amazon-Instanz-Metadaten

Amazon EC2 bietet einen Service für Instanzen mit dem Namen Instance Metadata Service (IMDS) an. Weitere Informationen zu diesem Service finden Sie unter [Arbeiten mit Instance-Metadaten](#) im EC2 Amazon-Benutzerhandbuch. Beim Versuch, Anmeldeinformationen auf einer EC2 Amazon-Instanz 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- geteilt AWS **config** Dateieinstellung,

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- geteilt AWS **config** Dateieinstellung,

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
```

Kompatibilität mit AWS SDKs

Im Folgenden werden die in diesem Thema beschriebenen Funktionen und Einstellungen SDKs unterstützt. Alle teilweisen Ausnahmen werden vermerkt. Alle Einstellungen für JVM Systemeigenschaften werden unterstützt von AWS SDK for Java und die AWS SDK for Kotlin nur.

SDK	U zt	Hinweise oder weitere Informationen
AWS CLI v2	Ja	
SDK für C++	Nein	
SDK für Go V2 (1.x)	Nein	
SDK für Go 1.x (V1)	Nein	
SDK für Java 2.x	Nein	
SDK für Java 1.x	Teilwe	Nur AWS_METADATA_SERVICE_TIMEOUT wird unterstüt
SDK für 3.x JavaScript	Nein	
SDK für 2.x JavaScript	Nein	
SDK für Kotlin	Nein	

SDK	U zt	Hinweise oder weitere Informationen
SDK für .NET 3.x	Nein	
SDK für 3.x PHP	Ja	
SDK für Python (Boto3)	Ja	
SDK für Ruby 3.x	Nein	
SDK für Rust	Nein	
SDK für Swift	Nein	
Tools für PowerShell	Nein	

Amazon S3 Access Points

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 den 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. für die Verwendung eines Access Points ARN mit [GetObject](#) ein Objekt aus einem Bucket abzurufen, oder für die Verwendung eines Access Points ARN mit [PutObject](#) ein Objekt zu einem Bucket hinzuzufügen.

Weitere Informationen zu Amazon S3 S3-Zugriffspunkten und ARNs finden Sie [unter Using Access Points](#) im Amazon S3 S3-Benutzerhandbuch.

Konfigurieren Sie diese Funktionalität wie folgt:

s3_use_arn_region- geteilt AWS **config**Dateieinstellung, **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 Ihren spezifischen Code. SDK

Diese Einstellung steuert, ob der den Access Point SDK verwendet ARN AWS-Region um den regionalen Endpunkt für die Anfrage zu erstellen. Das SDK bestätigt, dass ARN AWS-Region

wird von demselben serviert AWS Partition, wie der Client konfiguriert ist AWS-Region um partitionsübergreifende Aufrufe zu verhindern, die höchstwahrscheinlich fehlschlagen werden. Wenn mehrfach definiert, hat die vom Code konfigurierte Einstellung Vorrang, gefolgt von der Einstellung der Umgebungsvariablen.

Standardwert: `false`

Zulässige Werte:

- **true**— Der verwendet die SDK ARN AWS-Region beim Konstruieren des Endpunkts anstelle des vom Client konfigurierten AWS-Region. Ausnahme: Wenn der Client konfiguriert ist AWS-Region ist ein FIPS AWS-Region, dann muss es mit ARN den übereinstimmen AWS-Region. Andernfalls wird ein Fehler auftreten.
- **false**— Die SDK Nutzungen sind vom Client konfiguriert AWS-Region bei der Konstruktion des Endpunkts.

Kompatibilität mit AWS SDKs

Im Folgenden werden die in diesem Thema beschriebenen Funktionen und Einstellungen SDKs unterstützt. Alle teilweisen Ausnahmen werden vermerkt. Alle Einstellungen für JVM Systemeigenschaften werden unterstützt von AWS SDK for Java und die AWS SDK for Kotlin nur.

SDK	U zt	Notizen oder weitere Informationen
AWS CLI v2	Ja	
SDK für C++	Ja	
SDK für Go V2 (1.x)	Ja	
SDK für Go 1.x (V1)	Ja	Um die Einstellungen für gemeinsam genutzte <code>config</code> Dateien zu verwenden, müssen Sie das Laden aus der Konfigurationsdatei aktivieren. Weitere Informationen finden Sie unter Sessions .
SDK für Java 2.x	Ja	
SDK für Java 1.x	Ja	JVM Systemeigenschaft wird nicht unterstützt.

SDK	U zt	Notizen oder weitere Informationen
SDK für JavaScript 3.x	Ja	
SDK für 2.x JavaScript	Ja	
SDK für Kotlin	Ja	
SDK für .NET 3.x	Ja	Entspricht nicht der Standardpriorität; der Wert einer gemeinsam genutzten config Datei hat Vorrang vor der Umgebungsvariablen.
SDK für 3.x PHP	Ja	
SDK für Python (Boto3)	Ja	
SDK für Ruby 3.x	Ja	
SDK für Rust	Nein	
SDK für Swift	Nein	
Tools für PowerShell	Ja	Entspricht nicht der Standardpriorität; der Wert einer gemeinsam genutzten config Datei hat Vorrang vor der Umgebungsvariablen.

Multiregionale Amazon-S3-Zugriffspunkte

Amazon S3 Multiregion Access Points bieten einen globalen Endpunkt, über den Anwendungen Anfragen von Amazon S3 S3-Buckets bearbeiten können, die sich in mehreren AWS-Regionen. Sie können Multi-Region-Access Points verwenden, um multiregionale Anwendungen 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-Zugriffspunkte in Amazon S3](#) im Amazon S3-Benutzerhandbuch.

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

Weitere Informationen zum Erstellen von Access Points mit mehreren Regionen finden Sie unter [Verwaltung von Access Points mit mehreren Regionen](#) im Amazon S3 S3-Benutzerhandbuch.

Der Sigv4A-Algorithmus ist die Signaturimplementierung, die zum Signieren der globalen Regionsanfragen verwendet wird. Dieser Algorithmus wird SDK durch eine Abhängigkeit von der erhalten. [AWS Allgemeine Runtime \(CRT\) -Bibliotheken](#)

Konfigurieren Sie diese Funktionalität wie folgt:

s3_disable_multiregion_access_points- geteilt AWS **config**Dateieinstellung,
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 Ihren spezifischen Code. SDK

Diese Einstellung steuert, ob SDK potenziell regionsübergreifende Anfragen versucht werden. 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 multiregionalen Access Points.

Kompatibilität mit AWS SDKs

Im Folgenden werden die in diesem Thema beschriebenen Funktionen und Einstellungen SDKs unterstützt. Alle teilweisen Ausnahmen werden vermerkt. Alle Einstellungen für JVM Systemeigenschaften werden unterstützt von AWS SDK for Java und die AWS SDK for Kotlin nur.

SDK	U zt	Hinweise oder weitere Informationen
AWS CLI v2	Ja	

SDK	U zt	Hinweise oder weitere Informationen
SDK für C++	Ja	
SDK für Go V2 (1.x)	Ja	
SDK für Go 1.x (V1)	Nein	
SDK für Java 2.x	Ja	
SDK für Java 1.x	Nein	
SDK für 3.x JavaScript	Ja	
SDK für 2.x JavaScript	Nein	
SDK für Kotlin	Ja	
SDK für .NET 3.x	Ja	
SDK für 3.x PHP	Ja	
SDK für Python (Boto3)	Ja	
SDK für Ruby 3.x	Ja	
SDK für Rust	Ja	
SDK für Swift	Nein	
Tools für PowerShell	Ja	

AWS-Region

AWS-Regionen sind ein wichtiges Konzept, das man verstehen muss, wenn man damit arbeitet AWS-Services.

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. Amazon S3 und Amazon EC2 unterstützen beispielsweise die regionsübergreifende Replikation. Einige Dienste, wie 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.

Serviceclients erstellen

SDKsVerwenden Sie für den programmgesteuerten Zugriff AWS-Services jeweils eine Clientklasse/ ein Client-Objekt. AWS-Service Wenn Ihre Anwendung beispielsweise auf Amazon zugreifen mussEC2, würde Ihre Anwendung ein EC2 Amazon-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 `region` Einstellung festgelegt ist. 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 Sie angegeben SDK (überprüfen Sie Ihren spezifischen SDK Guide oder Ihre SDK Codebasis).

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 an, der für Anfragen verwendet AWS-Region werden soll. AWS Diese Region wird für SDK Serviceanfragen verwendet, für die keine bestimmte Region 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 „Konfigurationsobjekt“, mit dem die Standardregion im Anwendungscode festgelegt werden kann. Einzelheiten finden Sie in Ihrem speziellen AWS SDK Entwicklerhandbuch.

Kompatibilität mit AWS SDKs

Im Folgenden werden die in diesem Thema beschriebenen Funktionen und Einstellungen SDKs unterstützt. Alle teilweisen Ausnahmen werden vermerkt. Alle Einstellungen für JVM Systemeigenschaften werden AWS SDK for Kotlin nur von AWS SDK for Java und vom unterstützt.

SDK	U zt	Hinweise oder weitere Informationen
AWS CLI v2	Ja	AWS CLI v2 verwendet einen beliebigen Wert in <code>AWS_REGION</code> vor einem beliebigen Wert in <code>AWS_DEFAULT_REGION</code> (beide Variablen sind geprüft).
AWS CLI v1	Ja	AWS CLI v1 verwendet eine zu diesem <code>AWS_DEFAULT_REGION</code> Zweck benannte Umgebungsvariable.
SDK für C++	Ja	
SDK für Go V2 (1.x)	Ja	
SDK für Go 1.x (V1)	Ja	Um die Einstellungen für gemeinsam genutzte <code>config</code> Dateien zu verwenden, müssen Sie das Laden aus der Konfigurationsdatei aktivieren. Weitere Informationen finden Sie unter Sessions .
SDK für Java 2.x	Ja	
SDK für Java 1.x	Ja	
SDK für 3.x JavaScript	Ja	
SDK für 2.x JavaScript	Ja	
SDK für Kotlin	Ja	
SDK für .NET 3.x	Ja	
SDK für 3.x PHP	Ja	
SDK für Python (Boto3)	Ja	Dies SDK verwendet eine Umgebungsvariable, die zu diesem <code>AWS_DEFAULT_REGION</code> Zweck benannt wurde.
SDK für Ruby 3.x	Ja	
SDK für Rust	Ja	

SDK	U zt	Hinweise oder weitere Informationen
SDK für Swift	Ja	
Tools für PowerShell	Ja	

AWS STS Regionale Endpunkte

AWS Security Token Service (AWS STS) ist sowohl als globaler als auch als regionaler Service verfügbar. Einige von AWS SDKs und CLIs verwenden standardmäßig den globalen Dienstendpunkt (<https://sts.amazonaws.com>), während andere die regionalen Dienstendpunkte (https://sts.{region_identifizier}.{partition_domain}) verwenden. Globale Anfragen beziehen sich auf die Region USA Ost (Nord-Virginia). Weitere Informationen zu AWS STS Endpunkten finden Sie unter [Endpoints](#) in der AWS Security Token Service API-Referenz. Oder lernen Sie die [Verwaltung AWS STS in einem AWS-Region](#) im AWS Identity and Access Management Benutzerhandbuch kennen.

Es hat sich AWS bewährt, wann immer möglich regionale Endpunkte zu verwenden und Ihre [AWS-Region](#) zu konfigurieren. Kunden in anderen [Partitionen](#) als kommerziellen Partitionen müssen regionale Endpunkte verwenden. Nicht alle SDKs AND-Tools unterstützen diese Einstellung, aber alle haben ein definiertes Verhalten in Bezug auf globale und regionale Endpunkte. Weitere Informationen finden Sie im folgenden Abschnitt .

Für SDKs 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: legacy

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
```

Kompatibilität mit AWS SDKs

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` Dateivariablen 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	Standard- Einstellungswert	Standard-Ziel-STS- Endpunkt des Service- Clients	Fallback- Verhalten des Servicecl ients	Hinweise oder weitere Informationen
AWS CLI v2	Nein	N/A	Regionaler Endpunkt	Globaler Endpunkt
AWS CLI v1	Ja	legacy	Globaler Endpunkt	Globaler Endpunkt
SDK for C++	Nein	N/A	Regionaler Endpunkt	us-east-1 (Regional)
SDK for Go V2 (1.x)	Nein	N/A	Regionaler Endpunkt	Fehler bei der Anfrage

SDK	Standard-Instanzwert	Standard-Ziel-STS-Endpoint 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	Nein N/A	Regionaler Endpunkt	Fehler bei der Anfrage	Wenn keine Region konfiguriert ist, verwendet der AssumeRole und AssumeRoleWithWebIdentity den globalen STS-Endpoint.
SDK for Java 1.x	Ja legacy	Globaler Endpunkt	Globaler Endpunkt	
SDK für JavaScript 3.x	Nein N/A	Regionaler Endpunkt	Anfrage ist fehlgeschlagen	
SDK für JavaScript 2.x	Ja legacy	Globaler Endpunkt	Globaler Endpunkt	
SDK für Kotlin	Nein N/A	Regionaler Endpunkt	Globaler Endpunkt	
SDK for .NET 3.x	Ja legacy	Globaler Endpunkt	Globaler Endpunkt	

SDK	Standard- Einstellun- gswert	Standard- Ziel-STS- Endpunkt des Service- Clients	Fallback- Verhalten des Servicecl ients	Hinweise oder weitere Informationen
SDK for PHP 3.x	Ja legacy	Globaler Endpunkt	Fehler bei der Anfrage	
SDK for Python (Boto3)	Ja legacy	Globaler Endpunkt	Globaler Endpunkt	
SDK for Ruby 3.x	Ja regional	Regionaler Endpunkt	Anfrage ist fehlgesch lagen	
SDK für Rust	Nein N/A	Regionaler Endpunkt	Fehler bei der Anfrage	
SDK für Swift	Nein N/A	Regionaler Endpunkt	Fehler bei der Anfrage	
Tools für PowerShell	Ja legacy	Globaler Endpunkt	Globaler Endpunkt	

Schutz der Datenintegrität für Amazon S3

Ich unterstütze AWS SDKs seit einiger Zeit 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-basierte Algorithmen wie CRC32 NVME verwendet. CRC64 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 [Cyclic Redundancy Check \(CRC\) basierende Prüfsumme](#) für jeden Upload und sendet 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 SDKs berechnen sie 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 SDK-spezifischen Standardalgorithmus in der Tabelle validiert. [Kompatibilität mit AWS SDKs](#) Sie können sich für eine vollständige Objektprüfsumme entscheiden, indem Sie die Einstellung `checksum_type` auf `FULL_OBJECT` oder den NVME-Algorithmus verwenden. CRC64

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 CRC64 NVME-Prüfsumme für neue Objekte und speichert sie in den Objektmetadaten, damit Sie später darauf zugreifen können. Später können Sie den gespeicherten CRC 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](#)-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-spezifischer Standardalgorithmus verwendet. In der [Kompatibilität mit AWS SDKs](#) Tabelle finden Sie den Standardalgorithmus für jedes SDK oder Tool.

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 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 sind Benutzer beim Senden einer Anfrage für eine 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.

Kompatibilität mit AWS SDKs

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

Note

In der folgenden Tabelle bezieht sich „CRT“ auf das [AWS Allgemeine Runtime \(CRT\) - Bibliotheken](#) und erfordert möglicherweise das Hinzufügen einer zusätzlichen Abhängigkeit zu Ihrem Projekt.

SDK	Unterstützt	Standardmäßiger Prüfsummenalgorithmus	Unterstützte Prüfsummenalgorithmen	Hinweise oder weitere Informationen
AWS CLI v2	Ja	CRC64NVME	CRC64NVME, CRC32 C CRC32, SHA1 SHA256	Für AWS CLI Version 1 sind der Standardalgorithmus und die unterstützten Algorithmen identisch mit Python (Boto3).
SDK for C++	Ja	CRC64NVME	CRC64NVME, CRC32 C CRC32, SHA1 SHA256	
SDK for Go V2 (1.x)	Ja	CRC32	CRC32, CRC32 C,, SHA1 SHA256	

SDK	Unterstützt	Standardmäßiger Prüfsummenalgorithmus	Unterstützte Prüfsummenalgorithmen	Hinweise oder weitere Informationen
SDK for Go 1.x (V1)	Nein			
SDK for Java 2.x	Ja	CRC32	CRC64NVME (nur über CRT), C, CRC32, CRC32 SHA1 SHA256	
SDK for Java 1.x	Nein			
SDK für 3.x JavaScript	Ja	CRC32	CRC32, CRC32 C, SHA1 SHA256	
SDK für JavaScript 2.x	Nein			
SDK für Kotlin	Ja	CRC32	CRC32, CRC32 C, SHA1 SHA256	
SDK for .NET 3.x	Ja	CRC32	CRC32, CRC32 C, SHA1 SHA256	
SDK for PHP 3.x	Ja	CRC32	CRC32, CRC32 C (nur über CRT),, SHA1 SHA256	awscliFür die Verwendung CRC32 von C ist eine Erweiterung erforderlich.
SDK for Python (Boto3)	Ja	CRC32	CRC64NVME (nur über CRT) CRC32, CRC32 C (nur über CRT),, SHA1 SHA256	

SDK	Unterstützt	Standardmäßiger Prüfsummenalgorithmus	Unterstützte Prüfsummenalgorithmen	Hinweise oder weitere Informationen
SDK for Ruby 3.x	Ja	CRC32	CRC64NVME (nur über CRT) CRC32, CRC32 C (nur über CRT),, SHA1 SHA256	
SDK für Rust	Ja	CRC32	CRC64NVME CRC32, CRC32 C, SHA1 SHA256	
SDK für Swift	Ja	CRC32	CRC64NVME CRC32, CRC32 C, SHA1 SHA256	CRT-Abhängigkeit ist für alle Algorithmen erforderlich.
Tools für PowerShell	Ja	CRC32	CRC32, CRC32 C SHA1, SHA256	

Dual-Stack und Endpunkte FIPS

Konfigurieren Sie diese Funktionalität wie folgt:

use_dualstack_endpoint- geteilt AWS **config**Dateieinstellung,
AWS_USE_DUALSTACK_ENDPOINT- Umgebungsvariable, **aws.useDualstackEndpoint**- JVM Systemeigenschaft: Nur Java/Kotlin

Schaltet ein oder aus, ob Anfragen an Dual-Stack-Endpunkte gesendet SDK werden. Weitere Informationen zu Dual-Stack-Endpunkten, die IPv4 sowohl IPv6 Datenverkehr als auch unterstützen, finden Sie unter [Verwenden von Amazon S3 S3-Dual-Stack-Endpunkten](#) im Amazon Simple Storage Service-Benutzerhandbuch. Dual-Stack-Endpunkte sind für einige Services in einigen Regionen verfügbar.

Standardwert: `false`

Zulässige Werte:

- **true**— Das Tool SDK oder versucht, Dual-Stack-Endpunkte für Netzwerkanfragen zu verwenden. Wenn kein Dual-Stack-Endpunkt für den Dienst existiert und/oder AWS-Region, wird die Anfrage fehlschlagen.
- **false**— Das Tool SDK oder verwendet keine Dual-Stack-Endpunkte, um Netzwerkanfragen zu stellen.

use_fips_endpoint- gemeinsam genutzt AWS **config**Dateieinstellung,

AWS_USE_FIPS_ENDPOINT- Umgebungsvariable, **aws.useFipsEndpoint**- JVM

Systemeigenschaft: Nur Java/Kotlin

Schaltet ein oder aus, ob das Tool SDK oder Anfragen an -konforme Endpunkte sendet. FIPS 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 zum Standard AWS FIPSEndpunkte verwenden eine TLS Softwarebibliothek, die 140-2 entsprichtFIPS. Wenn diese Einstellung aktiviert ist und kein FIPS Endpunkt für den Dienst in Ihrem AWS-Region, der AWS Der Anruf kann fehlschlagen. [Servicespezifische Endpunkte](#)und die `--endpoint-url` Option für AWS Command Line Interface überschreibt diese Einstellung.

Um mehr über andere Möglichkeiten zur Angabe von FIPS Endpunkten zu erfahren, verwenden Sie AWS-Region, siehe [FIPSEndpunkte nach Dienst](#). Weitere Informationen zu Amazon Elastic Compute Cloud-Service-Endpunkten finden Sie unter [Dual-Stack IPv4 - \(undIPv6\) Endpoints](#) in der Amazon-Referenz. EC2 API

Standardwert: `false`

Zulässige Werte:

- **true**— Das Tool SDK oder sendet Anfragen an -konforme Endpunkte. FIPS
- **false**— Das Tool SDK oder sendet keine Anfragen an FIPS -konforme Endpunkte.

Kompatibilität mit AWS SDKs

Im Folgenden werden die in diesem Thema beschriebenen Funktionen und Einstellungen SDKs unterstützt. Alle teilweisen Ausnahmen werden vermerkt. Alle Einstellungen für JVM Systemeigenschaften werden unterstützt von AWS SDK for Java und die AWS SDK for Kotlin nur.

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

Endpunkterkennung

Verwenden Sie Endpoint Discovery für den Zugriff auf Dienstendpunkte (URLs für den Zugriff auf verschiedene Ressourcen) und behalten Sie gleichzeitig die Flexibilität für AWS nach URLs 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 SDK stellt der beispielsweise eine `DescribeEndpoints` Anfrage zum Abrufen der verfügbaren Endpunkte und verwendet diese Endpunkte dann, um bestimmte Operationen wie `createDatabase` oder `createTable` abzuschließen.

Konfigurieren Sie diese Funktionalität wie folgt:

endpoint_discovery_enabled- geteilt AWS `config` Dateieinstellung, **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 Ihren spezifischen Code. 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**— Der SDK sollte automatisch versuchen, einen Endpunkt für Dienste zu finden, bei denen die Endpunkterkennung optional ist.
- **false**— Der SDK sollte nicht automatisch versuchen, einen Endpunkt für Dienste zu finden, bei denen die Endpunkterkennung optional ist.

Kompatibilität mit AWS SDKs

Im Folgenden werden die in diesem Thema beschriebenen Funktionen und Einstellungen SDKs unterstützt. Alle teilweisen Ausnahmen werden vermerkt. Alle Einstellungen für JVM Systemeigenschaften werden unterstützt von AWS SDK for Java und die AWS SDK for Kotlin nur.

SDK	U zt	Notizen oder weitere Informationen
AWS CLI v2	Ja	
SDK für C++	Ja	
SDK für Go V2 (1.x)	Ja	
SDK für 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 für Java 2.x	Ja	Der SDK für Java 2.x verwendet <code>AWS_ENDPOINT_DISCOVERY_ENABLED</code> für die Umgebungsvariable den Namen.
SDK für Java 1.x	Teilwe	JVMSystemeigenschaft wird nicht unterstützt.
SDK für JavaScript 3.x	Ja	
SDK für 2.x JavaScript	Ja	
SDK für Kotlin	Ja	
SDK für .NET 3.x	Ja	
SDK für 3.x PHP	Ja	
SDK für Python (Boto3)	Ja	
SDK für Ruby 3.x	Ja	
SDK für Rust	Teilwe	Wird nur für Timestream unterstützt.
SDK für Swift	Neir	
Tools für PowerShell	Ja	

Allgemeine Konfigurationseinstellungen

SDKs unterstützt einige allgemeine Einstellungen, die das allgemeine SDK Verhalten konfigurieren.

Konfigurieren Sie diese Funktionalität wie folgt:

api_versions- geteilt AWS **config**Dateieinstellung

Etwas AWS Dienste verwalten mehrere API Versionen, um die Abwärtskompatibilität zu unterstützen. Standardmäßig und SDK AWS CLI Operationen verwenden die neueste verfügbare API Version. Wenn Sie eine bestimmte API Version für Ihre Anfragen benötigen möchten, nehmen Sie die `api_versions` Einstellung in Ihr Profil auf.

Standardwert: Keiner. (Die neueste API Version wird von der verwendet SDK.)

Gültige Werte: Dies ist eine verschachtelte Einstellung, auf die eine oder mehrere eingerückte Zeilen folgen, die jeweils eine Zeile kennzeichnen AWS Dienst und die zu API verwendende Version. Weitere Informationen finden Sie in der Dokumentation für AWS Service, um zu erfahren, welche API Versionen verfügbar sind.

Das Beispiel legt eine bestimmte API Version für zwei fest AWS Dienste in der `config` Datei. Diese API Versionen werden nur für Befehle verwendet, die unter dem Profil ausgeführt werden, das diese Einstellungen enthält. Befehle für jeden anderen Dienst verwenden die neueste Version dieses Dienstes API.

```
api_versions =  
  ec2 = 2015-03-01  
  cloudfront = 2015-09-017
```

ca_bundle- geteilt AWS **config**Dateieinstellung, **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- gemeinsam genutzt AWS **config**Dateieinstellung

Gibt an, wie Ergebnisse formatiert werden in AWS CLI und andere AWS SDKs und Werkzeuge.

Standardwert: `json`

Zulässige Werte:

- **[json](#)**— Die Ausgabe ist als [JSON](#) Zeichenfolge formatiert.
- **[yaml](#)**— Die Ausgabe ist als Zeichenfolge formatiert. [YAML](#)
- **[yaml-stream](#)**— Die Ausgabe wird gestreamt und als Zeichenfolge formatiert. [YAML](#) 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- geteilt AWS **config**Dateieinstellung

Gibt an, ob das Tool SDK oder versucht, Befehlszeilenparameter zu überprüfen, bevor es sie an die AWS Dienstendpunkt.

Standardwert: `true`

Zulässige Werte:

- **true** – Der Standardwert. Das Tool SDK oder führt eine clientseitige Überprüfung von Befehlszeilenparametern durch. Auf diese Weise kann das Tool SDK oder überprüfen, ob die Parameter gültig sind, und es werden einige Fehler erkannt. Das Tool SDK oder kann Anfragen zurückweisen, die nicht gültig sind, bevor es Anfragen an das AWS Dienstendpunkt.
- **false**— Das Tool SDK oder validiert Befehlszeilenparameter nicht, bevor es sie an die AWS Dienstendpunkt. Das Tool AWS Der Service-Endpunkt ist dafür verantwortlich, alle Anfragen zu validieren und Anfragen abzulehnen, die nicht gültig sind.

Kompatibilität mit AWS SDKs

Im Folgenden werden die in diesem Thema beschriebenen Funktionen und Einstellungen SDKs unterstützt. Alle teilweisen Ausnahmen werden vermerkt. Alle Einstellungen für JVM Systemeigenschaften werden unterstützt von AWS SDK for Java und die AWS SDK for Kotlin nur.

SDK	U zt	Notizen oder weitere Informationen
AWS CLI v2	Teilwe	<code>api_versions</code> nicht unterstützt.
SDK für C++	Ja	
SDK für Go V2 (1.x)	Teilwe	<code>api_versions</code> und wird <code>parameter_validation</code> nicht unterstützt.
SDK für Go 1.x (V1)	Teilwe	<code>api_versions</code> und wird <code>parameter_validation</code> nicht unterstützt. Um die Einstellungen für gemeinsam genutzte <code>config</code> Dateien zu verwenden, müssen Sie das Laden aus der Konfigurationsdatei aktivieren. Weitere Informationen finden Sie unter Sitzungen .

SDK	U zt	Notizen oder weitere Informationen
SDK für Java 2.x	Nein	
SDK für Java 1.x	Nein	
SDK für 3.x JavaScript	Ja	
SDK für 2.x JavaScript	Ja	
SDK für Kotlin	Nein	
SDK für .NET 3.x	Nein	
SDK für 3.x PHP	Ja	
SDK für Python (Boto3)	Ja	
SDK für Ruby 3.x	Ja	
SDK für Rust	Nein	
SDK für Swift	Nein	
Tools für PowerShell	Nein	

IMDSKlient

SDKs Implementieren Sie einen Instance-Metadaten-Service Version 2 (IMDSv2) -Client mithilfe von sitzungsorientierten Anfragen. Weitere Informationen zu IMDSv2 finden Sie unter [Verwendung IMDSv2](#) im EC2 Amazon-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

Das TTL des Tokens.

Standardwert: 21.600 Sekunden (6 Stunden, die maximal zugewiesene Zeit).

Gültige Werte: Zahl.

endpoint- Mitglied des Client-Konfigurationsobjekts

Der Endpunkt vonIMDS.

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

Gültige Werte: GültigURI.

Die folgenden Optionen werden von den meisten unterstützten SDKs. Einzelheiten finden Sie in Ihrer spezifischen SDK Codebasis.

endpoint_mode- Mitglied des Client-Konfigurationsobjekts

Der Endpunktmodus vonIMDS.

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 von. SDK

Gültige Werte: Variiert je nach SDK. Kann entweder ein numerischer Wert oder ein Aufruf einer benutzerdefinierten Funktion sein.

Kompatibilität mit AWS SDKs

Im Folgenden werden die in diesem Thema beschriebenen Funktionen und Einstellungen SDKs unterstützt. Alle teilweisen Ausnahmen werden vermerkt. Alle Einstellungen für JVM Systemeigenschaften werden AWS SDK for Kotlin nur von AWS SDK for Java und vom unterstützt.

SDK	U Hinweise oder weitere Informationen zt
AWS CLI v2	Ja
SDK für C++	Nein
SDK für Go V2 (1.x)	Ja
SDK für Go 1.x (V1)	Ja

SDK	U zt	Hinweise oder weitere Informationen
SDK für Java 2.x	Ja	
SDK für Java 1.x	Ja	
SDK für 3.x JavaScript	Ja	
SDK für 2.x JavaScript	Ja	
SDK für Kotlin	Nein	
SDK für .NET 3.x	Ja	
SDK für 3.x PHP	Ja	
SDK für Python (Boto3)	Ja	
SDK für Ruby 3.x	Ja	
SDK für Rust	Ja	
SDK für Swift	Ja	
Tools für PowerShell	Ja	

Wiederholungsverhalten

Das Wiederholungsverhalten umfasst Einstellungen, die festlegen, wie SDKs versucht wird, die Wiederherstellung nach Fehlern durchzuführen, die auf Anfragen zurückzuführen sind AWS-Services.

Konfigurieren Sie diese Funktionalität wie folgt:

retry_mode- geteilt AWS **config** Dateieinstellung, **AWS_RETRY_MODE**- Umgebungsvariable, **aws.retryMode**- JVM Systemeigenschaft: Nur Java/Kotlin

Gibt an, wie das SDK Entwicklertool versucht, es erneut zu versuchen.

Standardwert: Dieser Wert ist spezifisch für Ihren SDK. Suchen Sie in Ihrem spezifischen SDK Handbuch oder in Ihrer SDK Codebasis nach dem Standardwert `retry_mode`.

Zulässige Werte:

- `standard`— (Empfohlen) Der empfohlene Satz von Wiederholungsregeln für AWS SDKs. Dieser Modus umfasst eine Reihe von Standardfehlern, die wiederholt werden, und passt die Anzahl der Wiederholungsversuche automatisch an, um die Verfügbarkeit und Stabilität zu maximieren. Dieser Modus ist sicher für die Verwendung in Mehrmandantenanwendungen. Die standardmäßige maximale Anzahl von Versuchen in diesem Modus beträgt drei, sofern nicht `max_attempts` ausdrücklich konfiguriert.
- `adaptive`— Ein Wiederholungsmodus, der nur für spezielle Anwendungsfälle geeignet ist und die Funktionalität des Standardmodus sowie die automatische clientseitige Ratenbegrenzung umfasst. Dieser Wiederholungsmodus wird für Anwendungen mit mehreren Mandanten nicht empfohlen, es sei denn, Sie achten darauf, Anwendungsmandanten zu isolieren. Weitere Informationen finden Sie unter [Wählen Sie zwischen den Modi `standard` und `adaptive` versuchen Sie es erneut](#). Dieser Modus ist experimentell und könnte das Verhalten in future ändern.
- `legacy`— (Nicht empfohlen) Spezifisch für Sie SDK (überprüfen Sie Ihren spezifischen SDK Leitfaden oder Ihre SDK Codebasis).

max_attempts- geteilt AWS **config**Dateieinstellung, **AWS_MAX_ATTEMPTS**- Umgebungsvariable, **aws.maxAttempts**- JVM Systemeigenschaft: Nur Java/Kotlin

Gibt die maximale Anzahl an Versuchen an, die bei einer Anfrage unternommen werden können.

Standardwert: Wenn dieser Wert nicht angegeben ist, hängt sein Standardwert vom Wert der `retry_mode` Einstellung ab:

- Falls `retry_mode` ja `legacy` — Verwendet einen für `max_attempts` Sie spezifischen Standardwert SDK (den Standardwert finden Sie in Ihrer SDK spezifischen SDK Anleitung oder in Ihrer Codebasis).
- Falls `retry_mode` ja `standard` — Führt drei Versuche durch.
- Falls `retry_mode` ja `adaptive` — Führt drei Versuche durch.

Gültige Werte: Zahl größer als 0.

Wählen Sie zwischen den Modi **standard** und **adaptive** versuchen Sie es erneut

Wir empfehlen Ihnen, den `standard` Wiederholungsmodus zu verwenden, es sei denn, Sie sind sich sicher, dass Ihre Verwendung dafür besser geeignet ist. `adaptive`

Note

In diesem adaptive Modus wird davon ausgegangen, dass Sie Clients auf der Grundlage des Bereichs, in dem der Back-End-Dienst Anfragen drosseln kann, zusammenfassen. Wenn Sie dies nicht tun, können Drosselungen in einer Ressource Anfragen für eine nicht verwandte Ressource verzögern, wenn Sie denselben Client für beide Ressourcen verwenden.

Standard	Adaptiv
Anwendungsfälle: Alle.	Anwendungsfälle für Anwendungen: <ol style="list-style-type: none"> 1. Unempfindlich gegenüber Latenz. 2. Der Client greift nur auf eine einzelne Ressource zu, oder Sie stellen Logik bereit, um Ihre Clients getrennt nach der Dienstressource, auf die zugegriffen wird, in einem Pool zusammenzufassen.
Unterstützt Unterbrechungen, um zu verhindern, dass bei Ausfällen erneut SDK versucht wird.	Unterstützt das Unterbrechen von Stromkreisen, um zu verhindern, dass es bei Ausfällen erneut versucht. SDK
Verwendet bei Ausfällen einen exponentiellen Jitter-Backoff.	Verwendet dynamische Backoff-Dauern, um zu versuchen, die Anzahl der fehlgeschlagenen Anfragen zu minimieren, als Gegenleistung für die mögliche Erhöhung der Latenz.
Verzögert niemals den ersten Anforderungsversuch, sondern nur die Wiederholungsversuche.	Kann den ersten Anforderungsversuch drosseln oder verzögern.

Wenn Sie den adaptive Modus verwenden möchten, muss Ihre Anwendung Clients erstellen, die für jede Ressource konzipiert sind, die möglicherweise gedrosselt wird. In diesem Fall ist eine Ressource besser abgestimmt, als nur an jede einzelne Ressource zu denken AWS-Service. AWS-Services kann zusätzliche Dimensionen haben, die sie verwenden, um Anfragen zu drosseln. Lassen

Sie uns den Amazon DynamoDB-Service als Beispiel verwenden. DynamoDB verwendet AWS-Region plus die Tabelle, auf die zugegriffen wird, um Anfragen zu drosseln. Das bedeutet, dass eine Tabelle, auf die Ihr Code zugreift, möglicherweise stärker gedrosselt wird als andere. Wenn Ihr Code denselben Client für den Zugriff auf alle Tabellen verwendet hat und Anfragen an eine dieser Tabellen gedrosselt werden, reduziert der adaptive Wiederholungsmodus die Anforderungsrate für alle Tabellen. Ihr Code sollte so konzipiert sein, dass er einen Client pro Region-and-table R-Paar hat. Wenn Sie bei der Verwendung des adaptive Modus eine unerwartete Latenz feststellen, finden Sie weitere Informationen in den spezifischen AWS Dokumentationsleitfaden für den Dienst, den Sie verwenden.

Einzelheiten zur Implementierung im Wiederholungsmodus

Das Tool AWS SDKs verwenden Sie [Token-Buckets](#), um zu entscheiden, ob eine Anfrage erneut versucht werden soll und (im Fall des adaptive Wiederholungsmodus) wie schnell Anfragen gesendet werden sollen. Zwei Token-Buckets werden verwendet SDK: ein Token-Bucket für Wiederholungsversuche und ein Token-Bucket für die Anforderungsrate.

- Der Token-Bucket für Wiederholungen wird verwendet, um zu bestimmen, ob Wiederholungsversuche vorübergehend deaktiviert werden SDK sollen, um die Upstream- und Downstream-Dienste bei Ausfällen zu schützen. Token werden aus dem Bucket abgerufen, bevor Wiederholungsversuche unternommen werden, und Token werden an den Bucket zurückgegeben, wenn die Anfragen erfolgreich sind. Wenn der Bucket leer ist, wenn ein Wiederholungsversuch unternommen wird, SDK wird die Anfrage nicht erneut versucht.
- Der Token-Bucket für die Anforderungsrate wird nur im adaptive Wiederholungsmodus verwendet, um die Geschwindigkeit zu bestimmen, mit der Anfragen gesendet werden. Token werden aus dem Bucket abgerufen, bevor eine Anfrage gesendet wird, und Token werden mit einer dynamisch bestimmten Rate an den Bucket zurückgegeben, die auf Drosselungsantworten basiert, die vom Service zurückgegeben werden.

Im Folgenden finden Sie den allgemeinen Pseudocode für den Modus und den Wiederholungsmodus: standard adaptive

```
MakeSDKRequest() {
  attempts = 0
  loop {
    GetSendToken()
    response = SendHTTPRequest()
    RequestBookkeeping(response)
```

```
    if not Retryable(response)
        return response
    attempts += 1
    if attempts >= MAX_ATTEMPTS:
        return response
    if not HasRetryQuota(response)
        return response
    delay = ExponentialBackoff(attempts)
    sleep(delay)
}
```

Im Folgenden finden Sie weitere Informationen zu den im Pseudocode verwendeten Komponenten:

GetSendToken:

Dieser Schritt wird nur im adaptive Wiederholungsmodus verwendet. In diesem Schritt wird ein Token aus dem Token-Bucket für die Anforderungsrate abgerufen. Wenn ein Token nicht verfügbar ist, wartet es, bis eines verfügbar wird. SDKMöglicherweise stehen Ihnen Konfigurationsoptionen zur Verfügung, mit denen die Anfrage fehlschlagen kann, anstatt zu warten. Tokens im Bucket werden mit einer Geschwindigkeit aufgefüllt, die dynamisch auf der Grundlage der Anzahl der vom Client empfangenen Drosselungsantworten bestimmt wird.

SendHTTPRequest:

Dieser Schritt sendet die Anfrage an AWS. Die meisten AWS SDKs verwenden eine HTTP Bibliothek, die Verbindungspools verwendet, um eine bestehende Verbindung wiederzuverwenden, wenn HTTP Sie eine Anfrage stellen. Im Allgemeinen werden Verbindungen wiederverwendet, wenn eine Anfrage aufgrund von Drosselungsfehlern fehlgeschlagen ist, aber nicht, wenn eine Anfrage aufgrund eines vorübergehenden Fehlers fehlschlägt.

RequestBookkeeping:

Token werden dem Token-Bucket hinzugefügt, wenn die Anfrage erfolgreich ist. Nur im adaptive Wiederholungsmodus wird die Füllrate des Token-Buckets für die Anforderungsrate auf der Grundlage der Art der erhaltenen Antwort aktualisiert.

Retryable:

In diesem Schritt wird anhand der folgenden Kriterien bestimmt, ob eine Antwort erneut versucht werden kann:

- Der HTTP Statuscode.
- Der vom Dienst zurückgegebene Fehlercode.
- Verbindungsfehler, definiert als jeder Fehler, der vom Dienst empfangen wird und SDK bei dem keine HTTP Antwort vom Dienst empfangen wird.

Vorübergehende Fehler (HTTPStatuscodes 400, 408, 500, 502, 503 und 504) und Drosselungsfehler (HTTPStatuscodes 400, 403, 429, 502, 503 und 509) können alle potenziell wiederholt werden. SDKDas Wiederholungsverhalten wird in Kombination mit Fehlercodes oder anderen Daten aus dem Dienst bestimmt.

MAX_ATTEMPTS:

Die Standardanzahl der maximalen Versuche wird durch die `retry_mode` Einstellung festgelegt, sofern sie nicht durch die Einstellung überschrieben wird. `max_attempts`

HasRetryQuota

In diesem Schritt wird ein Token aus dem Token-Bucket für Wiederholungsversuche abgerufen. Wenn der Token-Bucket für Wiederholungen leer ist, wird die Anfrage nicht erneut versucht.

ExponentialBackoff

Bei einem Fehler, der erneut versucht werden kann, wird die Verzögerung beim erneuten Versuch anhand eines verkürzten exponentiellen Backoffs berechnet. Die SDKs Verwendung eines verkürzten binären exponentiellen Backoffs mit Jitter. Der folgende Algorithmus zeigt, wie die Zeit bis zum Schlafen (in Sekunden) für eine Antwort auf eine Anfrage definiert wird: i

```
seconds_to_sleep_i = min(b*r^i, MAX_BACKOFF)
```

Im vorherigen Algorithmus gelten die folgenden Werte:

b = random number within the range of: $0 \leq b \leq 1$

r = 2

`MAX_BACKOFF` = 20 seconds für die meisten SDKs. Weitere Informationen finden Sie in Ihrer spezifischen SDK Anleitung oder Ihrem Quellcode.

Kompatibilität mit AWS SDKs

Im Folgenden werden die in diesem Thema beschriebenen Funktionen und Einstellungen SDKs unterstützt. Alle teilweisen Ausnahmen werden vermerkt. Alle Einstellungen für JVM Systemeigenschaften werden unterstützt von AWS SDK for Java und die AWS SDK for Kotlin nur.

SDK	Unterstützt	Notizen oder weitere Informationen
AWS CLI v2	Ja	
SDK für C++	Ja	
SDK für Go V2 (1.x)	Ja	
SDK für Go 1.x (V1)	Nein	
SDK für Java 2.x	Ja	
SDK für Java 1.x	Ja	JVM Systemeigenschaften: <code>com.amazonaws.sdk.maxAttempts</code> anstelle von <code>com.amazonaws.sdk.retryMode</code> verwenden; <code>com.amazonaws.sdk.retryMode</code> anstelle von <code>aws.retryMode</code> verwenden.
SDK für JavaScript 3.x	Ja	
SDK für 2.x JavaScript	Nein	Unterstützt eine maximale Anzahl von Wiederholungsversuchen, exponentielles Backoff mit Jitter und eine Option für eine benutzerdefinierte Methode für Backoffwiederholungen.
SDK für Kotlin	Ja	
SDK für .NET 3.x	Ja	
SDK für 3.x PHP	Ja	
SDK für Python (Boto3)	Ja	
SDK für Ruby 3.x	Ja	
SDK für Rust	Ja	

SDK	U zt	Notizen oder weitere Informationen
SDK für Swift	Ja	
Tools für PowerShell	Ja	

Komprimierung anfordern

Note

Hilfe zum Verständnis des Layouts von Einstellungsseiten oder zur Interpretation der folgenden AWS SDKs Tabelle mit Kompatibilität finden Sie unter. [Seiten mit Einstellungen](#)

AWS SDKs und Tools können Payloads automatisch komprimieren, wenn Anfragen an diese gesendet werden AWS-Services, die den Empfang komprimierter Payloads unterstützen. 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 [PutMetricDataAPI-Methode](#).

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**— Schaltet die Anforderungskomprimierung aus.
- **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 des Anforderungstexts in Byte 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.

Kompatibilität mit AWS SDKs

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

SDK	U Hinweise oder weitere Informationen zt
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	U zt	Hinweise oder weitere Informationen
SDK für 3.x JavaScript	Ja	
SDK für 2.x JavaScript	Nein	
SDK für Kotlin	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	Ja	

Servicespezifische Endpunkte

Die dienstspezifische Endpunktconfiguration 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 unter [Identifikatoren 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 Endpunktconfigurationen 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 der gemeinsam genutzten Datei **config**

Wird in der gemeinsam genutzten `config` Datei an verschiedenen Stellen für unterschiedliche Funktionen verwendet. `endpoint_url`

- `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 der dienstspezifische Endpunkt URLs für Amazon S3 und AWS Elastic Beanstalk im selben Profil eingerichtet wird. 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-Identifizier-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 verwenden B und in Ihrem Code Amazon anrufen EC2, 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 instanziiieren.
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.

Kompatibilität mit AWS SDKs

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

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

SDK	U zt	Hinweise oder weitere Informationen
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 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	Ja	

Identifikatoren für dienstspezifische Endpunkte

Informationen darüber, wie und wo Sie die Identifikatoren in der folgenden Tabelle verwenden können, finden Sie unter [Servicespezifische Endpunkte](#)

serviceId	D	AWS_ENDPOINT_URL_<SERVICE>	Umgebungsvariable
	Di nu üs fü di ge ge Di Al co		
AccessAnalyzer	ar	AWS_ENDPOINT_URL_ACCESSANALYZER	
Account	ar	AWS_ENDPOINT_URL_ACCOUNT	
ACM	ar	AWS_ENDPOINT_URL_ACM	
ACM PCA	ar	AWS_ENDPOINT_URL_ACM_PCA	
Alexa For Business	ar	AWS_ENDPOINT_URL_ALEXA_FOR_BUSINESS _l	
amp	ar	AWS_ENDPOINT_URL_AMP	
Amplify	ar	AWS_ENDPOINT_URL_AMPLIFY	
AmplifyBackend	ar cl	AWS_ENDPOINT_URL_AMPLIFYBACKEND	
AmplifyUIBuilder	ar bt	AWS_ENDPOINT_URL_AMPLIFYUIBUILDER	
API Gateway	ar ay	AWS_ENDPOINT_URL_API_GATEWAY	

serviceId	Dienst	AWS_ENDPOINT_URL_<SERVICE>	Umgebungsvariable
ApiGatewayManagem entApi	API Gateway	AWS_ENDPOINT_URL_APIGATEWAYMANAGEMENTAPI	
ApiGatewayV2	API Gateway	AWS_ENDPOINT_URL_APIGATEWAYV2	
AppConfig	Amazon AppConfig	AWS_ENDPOINT_URL_APPCONFIG	
AppConfigData	Amazon AppConfig	AWS_ENDPOINT_URL_APPCONFIGDATA	
AppFabric	Amazon AppFabric	AWS_ENDPOINT_URL_APPFABRIC	
Appflow	Amazon AppFlow	AWS_ENDPOINT_URL_APPFLOW	
AppIntegrations	Amazon AppIntegrations	AWS_ENDPOINT_URL_APPINTEGRATIONS	
Application Auto Scaling	Amazon Application Auto Scaling	AWS_ENDPOINT_URL_APPLICATION_AUTO_SCALING	

serviceId	D:	AWS_ENDPOINT_URL_<SERVICE>	Umgebungsvariable
Application Insights	a	AWS_ENDPOINT_URL_APPLICATION_INSIGHTS	
ApplicationCostProfiler	a	AWS_ENDPOINT_URL_APPLICATIONCOSTPROFILER	
App Mesh	a	AWS_ENDPOINT_URL_APP_MESH	
AppRunner	a	AWS_ENDPOINT_URL_APPRUNNER	
AppStream	a	AWS_ENDPOINT_URL_APPSTREAM	
AppSync	a	AWS_ENDPOINT_URL_APPS_SYNC	
ARC Zonal Shift	a:	AWS_ENDPOINT_URL_ARC_ZONAL_SHIFT	
Artifact	a:	AWS_ENDPOINT_URL_ARTIFACT	
Athena	a	AWS_ENDPOINT_URL_ATHENA	
AuditManager	a	AWS_ENDPOINT_URL_AUDITMANAGER	

serviceId	D	AWS_ENDPOINT_URL_<SERVICE>	Umgebungsvariable
Auto Scaling	ai	AWS_ENDPOINT_URL_AUTO_SCALING	
Auto Scaling Plans	ai	AWS_ENDPOINT_URL_AUTO_SCALING_PLANS	
b2bi	b:	AWS_ENDPOINT_URL_B2BI	
Backup	b:	AWS_ENDPOINT_URL_BACKUP	
Backup Gateway	b:	AWS_ENDPOINT_URL_BACKUP_GATEWAY	
BackupStorage	b:	AWS_ENDPOINT_URL_BACKUPSTORAGE	
Batch	b:	AWS_ENDPOINT_URL_BATCH	
BCM Data Exports	b:	AWS_ENDPOINT_URL_BCM_DATA_EXPORTS	
Bedrock	b:	AWS_ENDPOINT_URL_BEDROCK	
Bedrock Agent	b:	AWS_ENDPOINT_URL_BEDROCK_AGENT	

serviceId	D: AWS_ENDPOINT_URL_<SERVICE> Umgebungsvariable
Bedrock Agent Runtime	b: AWS_ENDPOINT_URL_BEDROCK_AGENT_RUNTIME
Bedrock Runtime	b: AWS_ENDPOINT_URL_BEDROCK_RUNTIME
billingconductor	b: AWS_ENDPOINT_URL_BILLINGCONDUCTOR
Braket	b: AWS_ENDPOINT_URL_BRAKET
Budgets	b: AWS_ENDPOINT_URL_BUDGETS
Cost Explorer	c: AWS_ENDPOINT_URL_COST_EXPLORER
chatbot	c: AWS_ENDPOINT_URL_CHATBOT
Chime	c: AWS_ENDPOINT_URL_CHIME
Chime SDK Identity	c: AWS_ENDPOINT_URL_CHIME_SDK_IDENTITY
Chime SDK Media Pipelines	c: AWS_ENDPOINT_URL_CHIME_SDK_MEDIA_PIPELINES

serviceId	D: AWS_ENDPOINT_URL_<SERVICE> Umgebungsvariable
Chime SDK Meetings	<code>cl AWS_ENDPOINT_URL_CHIME_SDK_MEETINGS</code>
Chime SDK Messaging	<code>cl AWS_ENDPOINT_URL_CHIME_SDK_MESSAGING</code>
Chime SDK Voice	<code>cl AWS_ENDPOINT_URL_CHIME_SDK_VOICE</code>
CleanRooms	<code>c: AWS_ENDPOINT_URL_CLEANROOMS</code>
CleanRoomsML	<code>c: AWS_ENDPOINT_URL_CLEANROOMSML</code>
Cloud9	<code>c: AWS_ENDPOINT_URL_CLOUD9</code>
CloudControl	<code>c: AWS_ENDPOINT_URL_CLOUDCONTROL</code>
CloudDirectory	<code>c: AWS_ENDPOINT_URL_CLOUDDIRECTORY</code>
CloudFormation	<code>c: AWS_ENDPOINT_URL_CLOUDFORMATION</code>

serviceId	Default endpoint URL	Umgebungsvariable
CloudFront	c: AWS_ENDPOINT_URL_CLOUDFRONT	
CloudFront KeyValueCollection	c: AWS_ENDPOINT_URL_CLOUDFRONT_KEYVALUESTORE	
CloudHSM	c: AWS_ENDPOINT_URL_CLOUDHSM	
CloudHSM V2	c: AWS_ENDPOINT_URL_CLOUDHSM_V2	
CloudSearch	c: AWS_ENDPOINT_URL_CLOUDSEARCH	
CloudSearch Domain	c: AWS_ENDPOINT_URL_CLOUDSEARCH_DOMAIN	
CloudTrail	c: AWS_ENDPOINT_URL_CLOUDTRAIL	
CloudTrail Data	c: AWS_ENDPOINT_URL_CLOUDTRAIL_DATA	
CloudWatch	c: AWS_ENDPOINT_URL_CLOUDWATCH	

serviceId	Default endpoint URL	Umgebungsvariable
<code>codeartifact</code>	<code>aws://codeartifact</code>	<code>AWS_ENDPOINT_URL_CODEARTIFACT</code>
<code>CodeBuild</code>	<code>aws://codebuild</code>	<code>AWS_ENDPOINT_URL_CODEBUILD</code>
<code>CodeCatalyst</code>	<code>aws://codecatalyst</code>	<code>AWS_ENDPOINT_URL_CODECATALYST</code>
<code>CodeCommit</code>	<code>aws://codecommit</code>	<code>AWS_ENDPOINT_URL_CODECOMMIT</code>
<code>CodeDeploy</code>	<code>aws://codedeploy</code>	<code>AWS_ENDPOINT_URL_CODEDEPLOY</code>
<code>CodeGuru Reviewer</code>	<code>aws://codeguru-reviewer</code>	<code>AWS_ENDPOINT_URL_CODEGURU_REVIEWER</code>
<code>CodeGuru Security</code>	<code>aws://codeguru-security</code>	<code>AWS_ENDPOINT_URL_CODEGURU_SECURITY</code>
<code>CodeGuruProfiler</code>	<code>aws://codeguru-profiler</code>	<code>AWS_ENDPOINT_URL_CODEGURUPROFILER</code>
<code>CodePipeline</code>	<code>aws://codepipeline</code>	<code>AWS_ENDPOINT_URL_CODEPIPELINE</code>
<code>CodeStar</code>	<code>aws://codestar</code>	<code>AWS_ENDPOINT_URL_CODESTAR</code>

serviceId	Definition	Umweltvariable
	Die Endpunkte für die AWS-SDKs werden über die Umgebungsvariable <code>AWS_ENDPOINT_URL_<SERVICE></code> definiert.	
CodeStar connections	<code>AWS_ENDPOINT_URL_CODESTAR_CONNECTIONS</code>	
codestar notifications	<code>AWS_ENDPOINT_URL_CODESTAR_NOTIFICATIONS</code>	
Cognito Identity	<code>AWS_ENDPOINT_URL_COGNITO_IDENTITY</code>	
Cognito Identity Provider	<code>AWS_ENDPOINT_URL_COGNITO_IDENTITY_PROVIDER</code>	
Cognito Sync	<code>AWS_ENDPOINT_URL_COGNITO_SYNC</code>	
Comprehend	<code>AWS_ENDPOINT_URL_COMPREHEND</code>	
ComprehendMedical	<code>AWS_ENDPOINT_URL_COMPREHENDMEDICAL</code>	
Compute Optimizer	<code>AWS_ENDPOINT_URL_COMPUTE_OPTIMIZER</code>	

serviceId	Default endpoint URL	Umgebungsvariable
Config Service	<code>AWS_ENDPOINT_URL_CONFIG_SERVICE</code>	
Connect	<code>AWS_ENDPOINT_URL_CONNECT</code>	
Connect Contact Lens	<code>AWS_ENDPOINT_URL_CONNECT_CONTACT_LENS</code>	
ConnectCampaigns	<code>AWS_ENDPOINT_URL_CONNECTCAMPAIGNS</code>	
ConnectCases	<code>AWS_ENDPOINT_URL_CONNECTCASES</code>	
ConnectParticipant	<code>AWS_ENDPOINT_URL_CONNECTPARTICIPANT</code>	
ControlTower	<code>AWS_ENDPOINT_URL_CONTROLTOWER</code>	
Cost Optimization Hub	<code>AWS_ENDPOINT_URL_COST_OPTIMIZATION_HUB</code>	

serviceId	D: AWS_ENDPOINT_URL_<SERVICE> Umgebungsvariable
Cost and Usage Report Service	c: AWS_ENDPOINT_URL_COST_AND_USAGE_REPO u: RT_SERVICE o: c:
Customer Profiles	c: AWS_ENDPOINT_URL_CUSTOMER_PROFILES p:
DataBrew	d: AWS_ENDPOINT_URL_DATABREW
DataExchange	d: AWS_ENDPOINT_URL_DATAEXCHANGE n:
Data Pipeline	d: AWS_ENDPOINT_URL_DATA_PIPELINE l:
DataSync	d: AWS_ENDPOINT_URL_DATASYNC
DataZone	d: AWS_ENDPOINT_URL_DATAZONE
DAX	d: AWS_ENDPOINT_URL_DAX
Detective	d: AWS_ENDPOINT_URL_DETECTIVE
Device Farm	d: AWS_ENDPOINT_URL_DEVICE_FARM ir

serviceId	D:	Umgebungsvariable
	<code>AWS_ENDPOINT_URL_<SERVICE></code>	
DevOps Guru	<code>AWS_ENDPOINT_URL_DEVOPS_GURU</code>	
Direct Connect	<code>AWS_ENDPOINT_URL_DIRECT_CONNECT</code>	
Application Discovery Service	<code>AWS_ENDPOINT_URL_APPLICATION_DISCOVERY_SERVICE</code>	
DLM	<code>AWS_ENDPOINT_URL_DLM</code>	
Database Migration Service	<code>AWS_ENDPOINT_URL_DATABASE_MIGRATION_SERVICE</code>	
DocDB	<code>AWS_ENDPOINT_URL_DOCDB</code>	
DocDB Elastic	<code>AWS_ENDPOINT_URL_DOCDB_ELASTIC</code>	
drs	<code>AWS_ENDPOINT_URL_DRS</code>	
Directory Service	<code>AWS_ENDPOINT_URL_DIRECTORY_SERVICE</code>	
DynamoDB	<code>AWS_ENDPOINT_URL_DYNAMODB</code>	

serviceId	Default endpoint URL	Environment variable
	<code>AWS_ENDPOINT_URL_<SERVICE></code>	Umgebungsvariable
DynamoDB Streams	<code>AWS_ENDPOINT_URL_DYNAMODB_STREAMS</code>	
EBS	<code>AWS_ENDPOINT_URL_EBS</code>	
EC2	<code>AWS_ENDPOINT_URL_EC2</code>	
EC2 Instance Connect	<code>AWS_ENDPOINT_URL_EC2_INSTANCE_CONNECT</code>	
ECR	<code>AWS_ENDPOINT_URL_ECR</code>	
ECR PUBLIC	<code>AWS_ENDPOINT_URL_ECR_PUBLIC</code>	
ECS	<code>AWS_ENDPOINT_URL_ECS</code>	
EFS	<code>AWS_ENDPOINT_URL_EFS</code>	
EKS	<code>AWS_ENDPOINT_URL_EKS</code>	
EKS Auth	<code>AWS_ENDPOINT_URL_EKS_AUTH</code>	
Elastic Inference	<code>AWS_ENDPOINT_URL_ELASTIC_INFERENCE</code>	

serviceId	Di	AWS_ENDPOINT_URL_<SERVICE>	Umgebungsvariable
ElastiCache	e:	AWS_ENDPOINT_URL_ELASTICACHE	
Elastic Beanstalk	e:	AWS_ENDPOINT_URL_ELASTIC_BEANSTALK	
Elastic Transcoder	e:	AWS_ENDPOINT_URL_ELASTIC_TRANSCODER	
Elastic Load Balancing	e:	AWS_ENDPOINT_URL_ELASTIC_LOAD_BALANCING	
Elastic Load Balancing v2	e:	AWS_ENDPOINT_URL_ELASTIC_LOAD_BALANCING_V2	
EMR	er	AWS_ENDPOINT_URL_EMR	
EMR containers	er	AWS_ENDPOINT_URL_EMR_CONTAINERS	
EMR Serverless	er	AWS_ENDPOINT_URL_EMR_SERVERLESS	

serviceId	D: AWS_ENDPOINT_URL_<SERVICE> n üs fü di ge ge D: A c	Umgebungsvariable
EntityResolution	e: AWS_ENDPOINT_URL_ENTITYRESOLUTION o:	
Elasticsearch Service	e: AWS_ENDPOINT_URL_ELASTICSEARCH_SERVICE a: i:	
EventBridge	e: AWS_ENDPOINT_URL_EVENTBRIDGE g:	
Evidently	e: AWS_ENDPOINT_URL_EVIDENTLY	
finspace	f: AWS_ENDPOINT_URL_FINSPLACE	
finspace data	f: AWS_ENDPOINT_URL_FINSPLACE_DATA d:	
Firehose	f: AWS_ENDPOINT_URL_FIREHOSE	
fis	f: AWS_ENDPOINT_URL_FIS	
FMS	fr AWS_ENDPOINT_URL_FMS	
forecast	fr AWS_ENDPOINT_URL_FORECAST	
forecastquery	fr AWS_ENDPOINT_URL_FORECASTQUERY u:	

serviceId	D: AWS_ENDPOINT_URL_<SERVICE> Umgebungsvariable
FraudDetector	f: AWS_ENDPOINT_URL_FRAUDETECTOR c:
FreeTier	f: AWS_ENDPOINT_URL_FREETIER
FSx	f: AWS_ENDPOINT_URL_FSX
GameLift	g: AWS_ENDPOINT_URL_GAMELIFT
Glacier	g: AWS_ENDPOINT_URL_GLACIER
Global Accelerator	g: AWS_ENDPOINT_URL_GLOBAL_ACCELERATOR c:
Glue	g: AWS_ENDPOINT_URL_GLUE
grafana	g: AWS_ENDPOINT_URL_GRAFANA
Greengrass	g: AWS_ENDPOINT_URL_GREENGRASS s:
GreengrassV2	g: AWS_ENDPOINT_URL_GREENGRASSV2 s:
GroundStation	g: AWS_ENDPOINT_URL_GROUNDSTATION t:

serviceId	Di nu üs fü di ge ge D: A/ co	AWS_ENDPOINT_URL_<SERVICE>	Umgebungsvariable
GuardDuty	gü	AWS_ENDPOINT_URL_GUARDDUTY	
Health	he	AWS_ENDPOINT_URL_HEALTH	
HealthLake	he e	AWS_ENDPOINT_URL_HEALTHLAKE	
Honeycode	he	AWS_ENDPOINT_URL_HONEYCODE	
IAM	iä	AWS_ENDPOINT_URL_IAM	
identitystore	iä te	AWS_ENDPOINT_URL_IDENTITYSTORE	
imagebuilder	ir de	AWS_ENDPOINT_URL_IMAGEBUILDER	
ImportExport	ir o:	AWS_ENDPOINT_URL_IMPORTEXPORT	
Inspector	iä	AWS_ENDPOINT_URL_INSPECTOR	
Inspector Scan	iä _:	AWS_ENDPOINT_URL_INSPECTOR_SCAN	

serviceId	Definition	Umgebungsvariable
	Die AWS-Endpoint-URL für den Dienst <code><SERVICE></code> ist die URL, die für die Kommunikation mit dem Dienst verwendet wird.	
Inspector2	inspector2	AWS_ENDPOINT_URL_INSPECTOR2
InternetMonitor	internetmonitor	AWS_ENDPOINT_URL_INTERNETMONITOR
IoT	iot	AWS_ENDPOINT_URL_IOT
IoT Data Plane	iotdataplane	AWS_ENDPOINT_URL_IOT_DATA_PLANE
IoT Jobs Data Plane	iotjobsdataplane	AWS_ENDPOINT_URL_IOT_JOBS_DATA_PLANE
IoT 1Click Devices Service	iot1clickdevices	AWS_ENDPOINT_URL_IOT_1CLICK_DEVICES_SERVICE
IoT 1Click Projects	iot1clickprojects	AWS_ENDPOINT_URL_IOT_1CLICK_PROJECTS
IoTAnalytics	iotanalytics	AWS_ENDPOINT_URL_IOTANALYTICS

serviceId	Default endpoint URL	Umgebungsvariable
IotDeviceAdvisor	<code>AWS_ENDPOINT_URL_IOTDEVICEADVISOR</code>	
IoT Events	<code>AWS_ENDPOINT_URL_IOT_EVENTS</code>	
IoT Events Data	<code>AWS_ENDPOINT_URL_IOT_EVENTS_DATA</code>	
IoTFleetHub	<code>AWS_ENDPOINT_URL_IOTFLEETHUB</code>	
IoTFleetWise	<code>AWS_ENDPOINT_URL_IOTFLEETWISE</code>	
IoTSecureTunneling	<code>AWS_ENDPOINT_URL_IOTSECURETUNNELING</code>	
IoTSiteWise	<code>AWS_ENDPOINT_URL_IOTSITWISE</code>	
IoTThingsGraph	<code>AWS_ENDPOINT_URL_IOTTHINGSGRAPH</code>	
IoTTwinMaker	<code>AWS_ENDPOINT_URL_IOTTWINMAKER</code>	

serviceId	Default endpoint URL	Umgebungsvariable
IoT Wireless	<code>aws.iotwireless.endpoint.url</code>	<code>AWS_ENDPOINT_URL_IOT_WIRELESS</code>
ivs	<code>aws.ivs.endpoint.url</code>	<code>AWS_ENDPOINT_URL_IVS</code>
IVS RealTime	<code>aws.ivsrealtime.endpoint.url</code>	<code>AWS_ENDPOINT_URL_IVS_REALTIME</code>
ivschat	<code>aws.ivschat.endpoint.url</code>	<code>AWS_ENDPOINT_URL_IVSCHAT</code>
Kafka	<code>aws.kafka.endpoint.url</code>	<code>AWS_ENDPOINT_URL_KAFKA</code>
KafkaConnect	<code>aws.kafkaconnect.endpoint.url</code>	<code>AWS_ENDPOINT_URL_KAFKACONNECT</code>
kendra	<code>aws.kendra.endpoint.url</code>	<code>AWS_ENDPOINT_URL_KENDRA</code>
Kendra Ranking	<code>aws.kendra.endpoint.url</code>	<code>AWS_ENDPOINT_URL_KENDRA_RANKING</code>
Keyspaces	<code>aws.keyspaces.endpoint.url</code>	<code>AWS_ENDPOINT_URL_KEYSPACES</code>
Kinesis	<code>aws.kinesis.endpoint.url</code>	<code>AWS_ENDPOINT_URL_KINESIS</code>

serviceId	D: AWS_ENDPOINT_URL_<SERVICE> Umgebungsvariable
Kinesis Video Archived Media	k: AWS_ENDPOINT_URL_KINESIS_VIDEO_ARCHIVED_MEDIA
Kinesis Video Media	k: AWS_ENDPOINT_URL_KINESIS_VIDEO_MEDIA
Kinesis Video Signaling	k: AWS_ENDPOINT_URL_KINESIS_VIDEO_SIGNALING
Kinesis Video WebRTC Storage	k: AWS_ENDPOINT_URL_KINESIS_VIDEO_WEBRTC_STORAGE
Kinesis Analytics	k: AWS_ENDPOINT_URL_KINESIS_ANALYTICS
Kinesis Analytics V2	k: AWS_ENDPOINT_URL_KINESIS_ANALYTICS_V2
Kinesis Video	k: AWS_ENDPOINT_URL_KINESIS_VIDEO

serviceId	D: AWS_ENDPOINT_URL_<SERVICE> Umgebungsvariable
KMS	kr AWS_ENDPOINT_URL_KMS
LakeFormation	l: AWS_ENDPOINT_URL_LAKEFORMATION t:
Lambda	l: AWS_ENDPOINT_URL_LAMBDA
Launch Wizard	l: AWS_ENDPOINT_URL_LAUNCH_WIZARD z:
Lex Model Building Service	l: AWS_ENDPOINT_URL_LEX_MODEL_BUILDING_ _I SERVICE _:
Lex Runtime Service	l: AWS_ENDPOINT_URL_LEX_RUNTIME_SERVICE m: e:
Lex Models V2	l: AWS_ENDPOINT_URL_LEX_MODELS_V2 s:
Lex Runtime V2	l: AWS_ENDPOINT_URL_LEX_RUNTIME_V2 m:
License Manager	l: AWS_ENDPOINT_URL_LICENSE_MANAGER a:

serviceId	Default endpoint URL	Umgebungsvariable
License Manager Linux Subscriptions	1: AWS_ENDPOINT_URL_LICENSE_MANAGER_LINUX_SUBSCRIPTIONS	
License Manager User Subscriptions	1: AWS_ENDPOINT_URL_LICENSE_MANAGER_USER_SUBSCRIPTIONS	
Lightsail	1: AWS_ENDPOINT_URL_LIGHTSAIL	
Location	1: AWS_ENDPOINT_URL_LOCATION	
CloudWatch Logs	1: AWS_ENDPOINT_URL_CLOUDWATCH_LOGS	
CloudWatch Logs	1: AWS_ENDPOINT_URL_CLOUDWATCH_LOGS	
LookoutEquipment	1: AWS_ENDPOINT_URL_LOOKOUTEQUIPMENT	
LookoutMetrics	1: AWS_ENDPOINT_URL_LOOKOUTMETRICS	

serviceId	Default endpoint URL	Umgebungsvariable
LookoutVision	LookoutVisionEndpoint	AWS_ENDPOINT_URL_LOOKOUTVISION
M2	M2Endpoint	AWS_ENDPOINT_URL_M2
Machine Learning	MachineLearningEndpoint	AWS_ENDPOINT_URL_MACHINE_LEARNING
Macie2	Macie2Endpoint	AWS_ENDPOINT_URL_MACIE2
ManagedBlockchain	ManagedBlockchainEndpoint	AWS_ENDPOINT_URL_MANAGEDBLOCKCHAIN
ManagedBlockchain Query	ManagedBlockchainQueryEndpoint	AWS_ENDPOINT_URL_MANAGEDBLOCKCHAIN_QUERY
Marketplace Agreement	MarketplaceAgreementEndpoint	AWS_ENDPOINT_URL_MARKETPLACE_AGREEMENT
Marketplace Catalog	MarketplaceCatalogEndpoint	AWS_ENDPOINT_URL_MARKETPLACE_CATALOG

serviceId	Dienstname	Umgebungsvariable
	Dienstname	Umgebungsvariable
Marketplace Deployment	marketplace-deployment	AWS_ENDPOINT_URL_MARKETPLACE_DEPLOYMENT
Marketplace Entitlement Service	marketplace-entitlement	AWS_ENDPOINT_URL_MARKETPLACE_ENTITLEMENT_SERVICE
Marketplace Commerce Analytics	marketplace-commerce-analytics	AWS_ENDPOINT_URL_MARKETPLACE_COMMERCE_ANALYTICS
MediaConnect	mediaconnect	AWS_ENDPOINT_URL_MEDIACONNECT
MediaConvert	mediaconvert	AWS_ENDPOINT_URL_MEDIACONVERT
MediaLive	medialive	AWS_ENDPOINT_URL_MEDIALIVE
MediaPackage	mediapackage	AWS_ENDPOINT_URL_MEDIAPACKAGE

serviceId	Di nu üs fü di ge ge D: A/ co	AWS_ENDPOINT_URL_<SERVICE> Umgebungsvariable
MediaPackage Vod	m: a:	AWS_ENDPOINT_URL_MEDIAPACKAGE_VOD
MediaPackageV2	m: a:	AWS_ENDPOINT_URL_MEDIAPACKAGEV2
MediaStore	m: e	AWS_ENDPOINT_URL_MEDIASTORE
MediaStore Data	m: e_	AWS_ENDPOINT_URL_MEDIASTORE_DATA
MediaTailor	m: o:	AWS_ENDPOINT_URL_MEDIATAILOR
Medical Imaging	m: m:	AWS_ENDPOINT_URL_MEDICAL_IMAGING
MemoryDB	m: e	AWS_ENDPOINT_URL_MEMORYDB
Marketplace Metering	m: c: n:	AWS_ENDPOINT_URL_MARKETPLACE_METERING
Migration Hub	m: _f	AWS_ENDPOINT_URL_MIGRATION_HUB
mgn	m:	AWS_ENDPOINT_URL_MGN

serviceId	Default endpoint URL	Umgebungsvariable
Migration Hub Refactor Spaces	m: AWS_ENDPOINT_URL_MIGRATION_HUB_REFAC _I TOR_SPACES	
MigrationHub Config	m: AWS_ENDPOINT_URL_MIGRATIONHUB_CONFIG h g	
MigrationHubOrches trator	m: AWS_ENDPOINT_URL_MIGRATIONHUBORCHESTRATOR h t:	
MigrationHubStrategy	m: AWS_ENDPOINT_URL_MIGRATIONHUBSTRATEGY h g)	
Mobile	m: AWS_ENDPOINT_URL_MOBILE	
mq	m: AWS_ENDPOINT_URL_MQ	
MTurk	m: AWS_ENDPOINT_URL_MTURK	
MWAA	m: AWS_ENDPOINT_URL_MWAA	
Neptune	n: AWS_ENDPOINT_URL_NEPTUNE	

serviceId	Di	AWS_ENDPOINT_URL_<SERVICE>	Umgebungsvariable
	n: ü: fü: di: ge: ge: D: A: c:		
Neptune Graph	n:	AWS_ENDPOINT_URL_NEPTUNE_GRAPH	
neptunedata	n: t:	AWS_ENDPOINT_URL_NEPTUNEDATA	
Network Firewall	n: i:	AWS_ENDPOINT_URL_NETWORK_FIREWALL	
NetworkManager	n: n:	AWS_ENDPOINT_URL_NETWORKMANAGER	
NetworkMonitor	n: n:	AWS_ENDPOINT_URL_NETWORKMONITOR	
nimble	n:	AWS_ENDPOINT_URL_NIMBLE	
OAM	o:	AWS_ENDPOINT_URL_OAM	
Omics	o:	AWS_ENDPOINT_URL_OMICS	
OpenSearch	o: h:	AWS_ENDPOINT_URL_OPENSEARCH	
OpenSearchServerless	o: h: s:	AWS_ENDPOINT_URL_OPENSEARCHSERVERLESS	

serviceId	Default endpoint URL	Umgebungsvariable
	<code>aws://<SERVICE></code>	
OpsWorks	<code>opsworks</code>	<code>AWS_ENDPOINT_URL_OPSWORKS</code>
OpsWorksCM	<code>opsworkscm</code>	<code>AWS_ENDPOINT_URL_OPSWORKSCM</code>
Organizations	<code>organizations</code>	<code>AWS_ENDPOINT_URL_ORGANIZATIONS</code>
OSIS	<code>osis</code>	<code>AWS_ENDPOINT_URL_OSIS</code>
Outposts	<code>outposts</code>	<code>AWS_ENDPOINT_URL_OUTPOSTS</code>
p8data	<code>p8data</code>	<code>AWS_ENDPOINT_URL_P8DATA</code>
p8data	<code>p8data</code>	<code>AWS_ENDPOINT_URL_P8DATA</code>
Panorama	<code>panorama</code>	<code>AWS_ENDPOINT_URL_PANORAMA</code>
Payment Cryptography	<code>payment-cryptography</code>	<code>AWS_ENDPOINT_URL_PAYMENT_CRYPTOGRAPHY</code>
Payment Cryptography Data	<code>payment-cryptography-data</code>	<code>AWS_ENDPOINT_URL_PAYMENT_CRYPTOGRAPHY_DATA</code>
Pca Connector Ad	<code>pca-connector-ad</code>	<code>AWS_ENDPOINT_URL_PCA_CONNECTOR_AD</code>

serviceId	Di	AWS_ENDPOINT_URL_<SERVICE>	Umgebungsvariable
Personalize	p:	AWS_ENDPOINT_URL_PERSONALIZE	
Personalize Events	p:	AWS_ENDPOINT_URL_PERSONALIZE_EVENTS	
Personalize Runtime	p:	AWS_ENDPOINT_URL_PERSONALIZE_RUNTIME	
PI	p:	AWS_ENDPOINT_URL_PI	
Pinpoint	p:	AWS_ENDPOINT_URL_PINPOINT	
Pinpoint Email	p:	AWS_ENDPOINT_URL_PINPOINT_EMAIL	
Pinpoint SMS Voice	p:	AWS_ENDPOINT_URL_PINPOINT_SMS_VOICE	
Pinpoint SMS Voice V2	p:	AWS_ENDPOINT_URL_PINPOINT_SMS_VOICE_V2	
Pipes	p:	AWS_ENDPOINT_URL_PIPES	

serviceId	Default endpoint URL	Umgebungsvariable
	<code>DefaultEndpointURL</code>	
Polly	<code>polypolly</code>	<code>AWS_ENDPOINT_URL_POLLY</code>
Pricing	<code>pricingpricing</code>	<code>AWS_ENDPOINT_URL_PRICING</code>
PrivateNetworks	<code>privatenetworksprivatenetworks</code>	<code>AWS_ENDPOINT_URL_PRIVATENETWORKS</code>
Proton	<code>protonproton</code>	<code>AWS_ENDPOINT_URL_PROTON</code>
QBusiness	<code>qbusinessqbusiness</code>	<code>AWS_ENDPOINT_URL_QBUSINESS</code>
QConnect	<code>qconnectqconnect</code>	<code>AWS_ENDPOINT_URL_QCONNECT</code>
QLDB	<code>qldbqldb</code>	<code>AWS_ENDPOINT_URL_QLDB</code>
QLDB Session	<code>qldb-sessionqldb-session</code>	<code>AWS_ENDPOINT_URL_QLDB_SESSION</code>
QuickSight	<code>quicksightquicksight</code>	<code>AWS_ENDPOINT_URL_QUICKSIGHT</code>
RAM	<code>ramram</code>	<code>AWS_ENDPOINT_URL_RAM</code>
rbin	<code>rbinrbin</code>	<code>AWS_ENDPOINT_URL_RBIN</code>
RDS	<code>rdsrds</code>	<code>AWS_ENDPOINT_URL_RDS</code>
RDS Data	<code>rds-data</code>	<code>AWS_ENDPOINT_URL_RDS_DATA</code>

serviceId	Direktlink	Umgebungsvariable
	AWS_ENDPOINT_URL_<SERVICE>	
Redshift	AWS_ENDPOINT_URL_REDSHIFT	
Redshift Data	AWS_ENDPOINT_URL_REDSHIFT_DATA	
Redshift Serverless	AWS_ENDPOINT_URL_REDSHIFT_SERVERLESS	
Rekognition	AWS_ENDPOINT_URL_REKOGNITION	
repostspace	AWS_ENDPOINT_URL_REPOSTSPACE	
resiliencehub	AWS_ENDPOINT_URL_RESILIENCEHUB	
Resource Explorer 2	AWS_ENDPOINT_URL_RESOURCE_EXPLORER_2	
Resource Groups	AWS_ENDPOINT_URL_RESOURCE_GROUPS	

serviceId	<p>Die AWS_ENDPOINT_URL_<SERVICE> Umgebungsvariable</p> <p>nutzt die für die jeweilige Service-ID definierte Umgebungsvariable.</p>
Resource Groups Tagging API	<p>Die Umgebungsvariable AWS_ENDPOINT_URL_RESOURCE_GROUPS_TAGGING_API wird verwendet.</p>
RoboMaker	<p>Die Umgebungsvariable AWS_ENDPOINT_URL_ROBOMAKER wird verwendet.</p>
RolesAnywhere	<p>Die Umgebungsvariable AWS_ENDPOINT_URL_ROLESANYPWHERE wird verwendet.</p>
Route 53	<p>Die Umgebungsvariable AWS_ENDPOINT_URL_ROUTE_53 wird verwendet.</p>
Route53 Recovery Cluster	<p>Die Umgebungsvariable AWS_ENDPOINT_URL_ROUTE53_RECOVERY_CLUSTER wird verwendet.</p>
Route53 Recovery Control Config	<p>Die Umgebungsvariable AWS_ENDPOINT_URL_ROUTE53_RECOVERY_CONTROL_CONFIG wird verwendet.</p>
Route53 Recovery Readiness	<p>Die Umgebungsvariable AWS_ENDPOINT_URL_ROUTE53_RECOVERY_READINESS wird verwendet.</p>

serviceId	Default endpoint URL	Umgebungsvariable
Route 53 Domains	<code>AWS_ENDPOINT_URL_ROUTE_53_DOMAINS</code>	
Route53Resolver	<code>AWS_ENDPOINT_URL_ROUTE53RESOLVER</code>	
RUM	<code>AWS_ENDPOINT_URL_RUM</code>	
S3	<code>AWS_ENDPOINT_URL_S3</code>	
S3 Control	<code>AWS_ENDPOINT_URL_S3_CONTROL</code>	
S3Outposts	<code>AWS_ENDPOINT_URL_S3OUTPOSTS</code>	
SageMaker	<code>AWS_ENDPOINT_URL_SAGEMAKER</code>	
SageMaker A2I Runtime	<code>AWS_ENDPOINT_URL_SAGEMAKER_A2I_RUNTIME</code>	
Sagemaker Edge	<code>AWS_ENDPOINT_URL_SAGEMAKER_EDGE</code>	

serviceId	Default endpoint URL	Umgebungsvariable
SageMaker FeatureStore Runtime	<code>AWS_ENDPOINT_URL_SAGEMAKER_FEATURESTORE_RUNTIME</code>	
SageMaker Geospatial	<code>AWS_ENDPOINT_URL_SAGEMAKER_GEOSPATIAL</code>	
SageMaker Metrics	<code>AWS_ENDPOINT_URL_SAGEMAKER_METRICS</code>	
SageMaker Runtime	<code>AWS_ENDPOINT_URL_SAGEMAKER_RUNTIME</code>	
savingsplans	<code>AWS_ENDPOINT_URL_SAVINGSPLANS</code>	
Scheduler	<code>AWS_ENDPOINT_URL_SCHEDULER</code>	
schemas	<code>AWS_ENDPOINT_URL_SCHEMAS</code>	
SimpleDB	<code>AWS_ENDPOINT_URL_SIMPLEDB</code>	
Secrets Manager	<code>AWS_ENDPOINT_URL_SECRETS_MANAGER</code>	

serviceId	Default endpoint URL	Umgebungsvariable
SecurityHub	SECURITYHUB	AWS_ENDPOINT_URL_SECURITYHUB
SecurityLake	SECURITYLAKE	AWS_ENDPOINT_URL_SECURITYLAKE
ServerlessApplicationRepository	SERVERLESSAPPLICATIONREPOSITORY	AWS_ENDPOINT_URL_SERVERLESSAPPLICATIONREPOSITORY
Service Quotas	SERVICEQUOTAS	AWS_ENDPOINT_URL_SERVICEQUOTAS
Service Catalog	SERVICECATALOG	AWS_ENDPOINT_URL_SERVICECATALOG
Service Catalog AppRegistry	SERVICECATALOGAPPREGISTRY	AWS_ENDPOINT_URL_SERVICECATALOGAPPREGISTRY
ServiceDiscovery	SERVICEDISCOVERY	AWS_ENDPOINT_URL_SERVICEDISCOVERY
SES	SES	AWS_ENDPOINT_URL_SES

serviceId	Default endpoint URL	Umgebungsvariable
SESV2	<code>aws.EndpointUrlSesv2</code>	<code>AWS_ENDPOINT_URL_SESV2</code>
Shield	<code>aws.EndpointUrlShield</code>	<code>AWS_ENDPOINT_URL_SHIELD</code>
signer	<code>aws.EndpointUrlSigner</code>	<code>AWS_ENDPOINT_URL_SIGNER</code>
SimSpaceWeaver	<code>aws.EndpointUrlSimSpaceWeaver</code>	<code>AWS_ENDPOINT_URL_SIMSPACEWEAVER</code>
SMS	<code>aws.EndpointUrlSms</code>	<code>AWS_ENDPOINT_URL_SMS</code>
Snow Device Management	<code>aws.EndpointUrlSnowDeviceManagement</code>	<code>AWS_ENDPOINT_URL_SNOW_DEVICE_MANAGEMENT</code>
Snowball	<code>aws.EndpointUrlSnowball</code>	<code>AWS_ENDPOINT_URL_SNOWBALL</code>
SNS	<code>aws.EndpointUrlSns</code>	<code>AWS_ENDPOINT_URL_SNS</code>
SQS	<code>aws.EndpointUrlSqs</code>	<code>AWS_ENDPOINT_URL_SQS</code>
SSM	<code>aws.EndpointUrlSsm</code>	<code>AWS_ENDPOINT_URL_SSM</code>
SSM Contacts	<code>aws.EndpointUrlSsmContacts</code>	<code>AWS_ENDPOINT_URL_SSM_CONTACTS</code>
SSM Incidents	<code>aws.EndpointUrlSsmIncidents</code>	<code>AWS_ENDPOINT_URL_SSM_INCIDENTS</code>

serviceId	D: AWS_ENDPOINT_URL_<SERVICE>	Umgebungsvariable
Ssm Sap	s: AWS_ENDPOINT_URL_SSM_SAP	
SSO	s: AWS_ENDPOINT_URL_SSO	
SSO Admin	s: AWS_ENDPOINT_URL_SSO_ADMIN	
SSO OIDC	s: AWS_ENDPOINT_URL_SSO_OIDC	
SFN	s: AWS_ENDPOINT_URL_SFN	
Storage Gateway	s: AWS_ENDPOINT_URL_STORAGE_GATEWAY	
STS	s: AWS_ENDPOINT_URL_STS	
SupplyChain	s: AWS_ENDPOINT_URL_SUPPLYCHAIN	
Support	s: AWS_ENDPOINT_URL_SUPPORT	
Support App	s: AWS_ENDPOINT_URL_SUPPORT_APP	
SWF	s: AWS_ENDPOINT_URL_SWF	
synthetics	s: AWS_ENDPOINT_URL_SYNTHETICS	

serviceId	D: AWS_ENDPOINT_URL_<SERVICE> Umgebungsvariable
Textract	t: AWS_ENDPOINT_URL_TEXTRACT
Timestream InfluxDB	t: AWS_ENDPOINT_URL_TIMESTREAM_INFLUXDB m_ b
Timestream Query	t: AWS_ENDPOINT_URL_TIMESTREAM_QUERY m_
Timestream Write	t: AWS_ENDPOINT_URL_TIMESTREAM_WRITE m_
tnb	t: AWS_ENDPOINT_URL_TNB
Transcribe	t: AWS_ENDPOINT_URL_TRANSCRIBE e
Transfer	t: AWS_ENDPOINT_URL_TRANSFER
Translate	t: AWS_ENDPOINT_URL_TRANSLATE
TrustedAdvisor	t: AWS_ENDPOINT_URL_TRUSTEDADVISOR v:
VerifiedPermissions	v: AWS_ENDPOINT_URL_VERIFIEDPERMISSIONS e: s

serviceId	D	AWS_ENDPOINT_URL_<SERVICE>	Umgebungsvariable
Voice ID	v	AWS_ENDPOINT_URL_VOICE_ID	
VPC Lattice	v	AWS_ENDPOINT_URL_VPC_LATTICE	
WAF	w	AWS_ENDPOINT_URL_WAF	
WAF Regional	w	AWS_ENDPOINT_URL_WAF_REGIONAL	
WAFV2	w	AWS_ENDPOINT_URL_WAFV2	
WellArchitected	w	AWS_ENDPOINT_URL_WELLARCHITECTED	
Wisdom	w	AWS_ENDPOINT_URL_WISDOM	
WorkDocs	w	AWS_ENDPOINT_URL_WORKDOCS	
WorkLink	w	AWS_ENDPOINT_URL_WORKLINK	
WorkMail	w	AWS_ENDPOINT_URL_WORKMAIL	
WorkMailMessageFlow	w	AWS_ENDPOINT_URL_WORKMAILMESSAGEFLOW	

serviceId	D: AWS_ENDPOINT_URL_<SERVICE> n üs fü di ge ge D: A/ c	Umgebungsvariable
WorkSpaces	w: AWS_ENDPOINT_URL_WORKSPACES S	
WorkSpaces Thin Client	w: AWS_ENDPOINT_URL_WORKSPACES_THIN_CLIENT S_ i	
WorkSpaces Web	w: AWS_ENDPOINT_URL_WORKSPACES_WEB S_	
XRay	x: AWS_ENDPOINT_URL_XRAY	

Standardeinstellungen für intelligente Konfigurationen

Mit der Funktion „Standardeinstellungen für intelligente Konfigurationen“ AWS SDK kann vordefinierte, optimierte Standardwerte für andere Konfigurationseinstellungen bereitstellen.

Konfigurieren Sie diese Funktionalität wie folgt:

defaults_mode- geteilt AWS **config**Dateieinstellung, **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 ein AWS SDK für eine Einstellung ist ein Wert explizit festgelegt, dann hat dieser Wert immer Vorrang. Wenn ein AWS SDK für eine Einstellung wurde kein explizit festgelegter Wert festgelegt und `defaults_mode` sie entspricht auch nicht der alten Einstellung, kann diese Funktion unterschiedliche

Standardwerte für verschiedene Einstellungen bereitstellen, die für Ihre Anwendung optimiert sind. Zu den Einstellungen können Folgendes gehören: HTTP Kommunikationseinstellungen, Wiederholungsverhalten, regionale Endpunkteinstellungen des Dienstes und möglicherweise jede zugehörige SDK 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`, beim Upgrade von Tests Ihrer Anwendung durchzuführen SDK, 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 zur Verfügung, die je nach Einrichtung von variieren SDK und vor der Einrichtung von existiert `defaults_mode`.
- `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 AWS-Region.
- `cross-region`— Baut auf dem Standardmodus auf und beinhaltet eine Optimierung, die auf Anwendungen zugeschnitten ist, die aufrufen AWS-Services in einer anderen Region.
- `mobile`— Baut auf dem Standardmodus auf und beinhaltet eine auf mobile Anwendungen zugeschnittene Optimierung.
- `auto`— Baut auf dem Standardmodus auf und beinhaltet experimentelle Funktionen. Die SDK Versuche, 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, `defaults_mode` stattdessen eine explizite Latenz 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 `defaults_mode` werdendefaults_mode:

- `retryMode`— Gibt an, wie die SDK Versuche wiederholt werden. Siehe [Wiederholungsverhalten](#).
- `stsRegionalEndpoints`— Gibt an, wie der SDK bestimmt AWS-Service Endpunkt, über den es mit dem kommuniziert AWS Security Token Service (AWS STS). Seht [AWS STS Regionale Endpunkte](#).
- `s3UsEast1RegionalEndpoints`— Gibt an, wie der SDK bestimmt AWS Service-Endpunkt, über den es mit Amazon S3 für die `us-east-1` Region kommuniziert.
- `connectTimeoutInMillis`— Nach einem ersten Verbindungsversuch auf einem Socket die Zeit 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 dauern 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 der für Ihre Anwendung `defaults_mode` ausgewählten Einstellung. Diese Werte sind derzeit wie folgt festgelegt (Änderungen vorbehalten):

Parameter	Modus standard	Modus in-region	Modus cross-region	Modus mobile
<code>retryMode</code>	<code>standard</code>	<code>standard</code>	<code>standard</code>	<code>standard</code>
<code>stsRegionalEndpoints</code>	<code>regional</code>	<code>regional</code>	<code>regional</code>	<code>regional</code>
<code>s3UsEast1RegionalEndpoints</code>	<code>regional</code>	<code>regional</code>	<code>regional</code>	<code>regional</code>
<code>connectTimeoutInMillis</code>	3100	1100	3100	30000

Parameter	Modus standard	Modus in-region	Modus cross-region	Modus mobile
<code>tlsNegotiationTimeoutInMillis</code>	3100	1100	3100	30000

Wenn `defaults_mode` Sie beispielsweise „`standard`“ ausgewählt haben, 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.

Kompatibilität mit AWS SDKs

Im Folgenden werden die in diesem Thema beschriebenen Funktionen und Einstellungen SDKs unterstützt. Alle teilweisen Ausnahmen werden vermerkt. Alle Einstellungen für JVM Systemeigenschaften werden unterstützt von AWS SDK for Java und die AWS SDK for Kotlin nur.

SDK	Unterstützt	Hinweise oder weitere Informationen
AWS CLI v2	Nein	
SDK für C++	Ja	Parameter nicht optimiert: <code>stsRegionalEndpoints</code> , <code>s3UsEast1RegionalEndpoints</code> , <code>tlsNegotiationTimeoutInMillis</code> .
SDK für Go V2 (1.x)	Ja	Parameter nicht optimiert: <code>retryMode</code> , <code>stsRegionalEndpoints</code> , <code>s3UsEast1RegionalEndpoints</code> .
SDK für Go 1.x (V1)	Nein	

SDK	Unterstützt	Hinweise oder weitere Informationen
SDK für Java 2.x	Ja	Parameter nicht optimiert: <code>stsRegionalEndpoints</code> .
SDK für Java 1.x	Nein	
SDK für 3.x JavaScript	Ja	Parameter nicht optimiert: <code>stsRegionalEndpoints</code> , <code>s3UsEast1RegionalEndpoints</code> , <code>tlsNegotiationTimeoutInMillis</code> . <code>connectTimeoutInMillis</code> heißt <code>connectionTimeout</code> .
SDK für JavaScript 2.x	Nein	
SDK für Kotlin	Nein	
SDK für .NET 3.x	Ja	Parameter nicht optimiert: <code>connectTimeoutInMillis</code> , <code>tlsNegotiationTimeoutInMillis</code> .
SDK für PHP 3.x	Ja	Parameter nicht optimiert: <code>tlsNegotiationTimeoutInMillis</code> .
SDK für Python (Boto3)	Ja	Parameter nicht optimiert: <code>tlsNegotiationTimeoutInMillis</code>
SDK für Ruby 3.x	Ja	

SDK	Unterstützt	Hinweise oder weitere Informationen
SDK für Rust	Nein	
SDK für Swift	Nein	
Tools für PowerShell	Ja	Parameter nicht optimiert :connectTimeoutInMillis ,tlsNegotiationTimeoutInMillis .

AWS Allgemeine Runtime (CRT) -Bibliotheken

Die AWS Common Runtime (CRT) -Bibliotheken sind eine Basisbibliothek von 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 gemeinsam und SDKs bieten 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,,), Unterzeichner, SHA256 SHA256 HMAC AES
- [awslabs/aws-c-common](#): Grundlegende Datenstrukturen, primitive Thread-/Synchronisationstypen, Pufferverwaltung, stdlib-bezogene Funktionen
- [awslabs/aws-c-compression](#): Komprimierungsalgorithmen (Huffman-Kodierung/Dekodierung)
- [awslabs/aws-c-event-stream](#): Verarbeitung von Event-Stream-Nachrichten (Header, Prelude, Payload, CRC/Trailer), Implementierung von Remote Procedure Call () über Event-Streams RPC
- [awslabs/aws-c-http](#): C99-Implementierung der /1.1- und /2-Spezifikationen HTTP HTTP
- [awslabs/aws-c-io](#): Sockets (TCP,UDP), PipesDNS, Event-Loops, Kanäle,/SSLTLS
- [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-Instances mit hoher Bandbreite EC2
- [awslabs/aws-c-sdkutils](#): Eine Dienstprogramm-Bibliothek zum Analysieren und Verwalten von Profilen AWS
- [awslabs/aws-checksums](#): Plattformübergreifend, hardwarebeschleunigt CRC32c und CRC32 mit Rückgriff auf effiziente Softwareimplementierungen
- [awslabs/aws-1c](#): Kryptografische Bibliothek für allgemeine Zwecke, die AWS vom Cryptography-Team AWS und seinen Kunden verwaltet wird und auf Code aus dem Google Boring-Projekt und dem Open-Projekt basiert SSL SSL

- [aws-labs/s2n](#): C99-Implementierung der TLS SSL /-Protokolle, die so konzipiert sind, dass sie klein und schnell sind, wobei Sicherheit an erster Stelle steht

Das CRT ist für alle SDKs außer Go und Rust verfügbar.

CRTAbhängigkeiten

Die CRT Bibliotheken bilden ein komplexes Netz von Beziehungen und Abhängigkeiten. Die Kenntnis dieser Beziehungen ist hilfreich, wenn Sie sie CRT direkt aus dem Quellcode erstellen müssen. Die meisten Benutzer greifen jedoch über ihre Sprache SDK (z. B. AWS SDK für C++ oder AWS SDK für Java) oder ihr Sprach-IoT-Gerät SDK (wie AWS IoT SDK für C++ oder AWS IoT SDK für Java) auf CRT Funktionen zu. In der folgenden Abbildung bezieht sich das Feld CRT Sprachbindungen auf das Paket, das die CRT Bibliotheken für eine bestimmte Sprache SDK umschließt. Dies ist eine Sammlung von Paketen in der Form `aws-crt-*`, wobei `*` für eine SDK Sprache steht (z. B. [aws-crt-cpp](#) oder [aws-crt-java](#)).

Im Folgenden werden die hierarchischen Abhängigkeiten der CRT Bibliotheken veranschaulicht.

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-Releases werden für Paketmanager (z. B. Maven NuGet, PyPI) veröffentlicht und sind als Quellcode verfügbar. [GitHub](#)

Wir empfehlen Benutzern, up-to-date bei SDK-Versionen zu bleiben, um über die neuesten Funktionen, Sicherheitsupdates und die zugrunde liegenden Abhängigkeiten auf dem Laufenden zu bleiben. Die fortgesetzte Verwendung einer SDK-Version, die nicht unterstützt wird, wird nicht empfohlen und erfolgt nach eigenem Ermessen des Benutzers.

Versionsverwaltung

Die AWS SDK-Release-Versionen haben die Form X.Y.Z, wobei 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 Early-Access-Zwecke und Feedback-Zwecke 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 regelmäßig Releases bereitstellen, die neue Funktionsupdates und Bugfixes beinhalten. 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.
- **Ende des Supports (Phase 4)** — Wenn ein SDK das Ende des Support erreicht, 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, das erreicht wurde, end-of-support erfolgt nach eigenem 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.

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 end-of-support Zeitplan 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 werden der Weg dazu beschrieben end-of-support, 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 zur end-of-support Kampagne skizziert und die Zeitpläne der Kampagne wiederholt.
- Den SDKs wurden Warnungen vor veralteten Versionen hinzugefügt, in denen der Pfad zur SDK-Dokumentation beschrieben und auf sie end-of-support verlinkt wird.

Eine Liste der verfügbaren Hauptversionen von AWS SDKs und Tools sowie deren Status im Wartungszyklus finden Sie unter. [Versionsunterstützung](#)

AWS SDKs und Tools-Versionsunterstützung

Die folgende Tabelle zeigt die Liste der verfügbaren AWS Hauptversionen des Software Development Kit (SDK) und deren Position im Wartungslebenszyklus mit zugehörigen Zeitplänen. Für detaillierte Informationen über den Lebenszyklus der Hauptversionen von AWS SDKs und Tools und die ihnen zugrunde liegenden Abhängigkeiten finden Sie unter [Wartungsrichtlinie](#).

SDK	Hauptversion	Aktuelle Phase	Datum der allgemeinen Verfügbarkeit	Hinweise
AWS CLI	1.x	Allgemeine Verfügbarkeit	9/2/2013	
AWS CLI	2.x	Allgemeine Verfügbarkeit	10.02.2020	
SDK für C++	1.x	Allgemeine Verfügbarkeit	9/2/2015	
SDK für Go V2	V2 1.x	Allgemeine Verfügbarkeit	19.1.2021	
SDK für Go	1.x	Wartung	19.11.2015	Einzelheiten und Termine finden Sie in der Ankündigung
SDK für Java	1.x	Wartung	25.03.2010	Einzelheiten und Termine finden Sie in der Ankündigung
SDK für Java	2.x	Allgemeine Verfügbarkeit	20.11.2018	
SDK für JavaScript	1.x	Ende des Supports	06.05.2013	

SDK	Hauptversion	Aktuelle Phase	Datum der allgemeinen Verfügbarkeit	Hinweise
SDK für JavaScript	2.x	Wartung	19.06.2014	Einzelheiten und Termine finden Sie in der Ankündigung
SDK für JavaScript	3.x	Allgemeine Verfügbarkeit	15.12.2020	
SDK für Kotlin	1.x	Allgemeine Verfügbarkeit	27.11.2023	
SDK für .NET	1.x	Ende des Supports	11/2009	
SDK für .NET	2.x	Ende des Supports	8.11.2013	
SDK für .NET	3.x	Allgemeine Verfügbarkeit	28.7.2015	
SDK für PHP	2.x	Ende des Supports	02.11.2012	
SDK für PHP	3.x	Allgemeine Verfügbarkeit	27.5.2015	
SDK für Python (Boto2)	1.x	Ende des Supports	13.07.2011	
SDK für Python (Boto3)	1.x	Allgemeine Verfügbarkeit	22.06.2015	
SDK für Python (Botocore)	1.x	Allgemeine Verfügbarkeit	22.06.2015	

SDK	Hauptversion	Aktuelle Phase	Datum der allgemeinen Verfügbarkeit	Hinweise
SDK für Ruby	1.x	Ende des Supports	14.7.2011	
SDK für Ruby	2.x	Ende des Supports	15.02.2015	
SDK für Ruby	3.x	Allgemeine Verfügbarkeit	29.8.2017	
SDK für Rust	1.x	Allgemeine Verfügbarkeit	27.11.2023	
SDK für Swift	1.x	Allgemeine Verfügbarkeit	17.9.2024	
Werkzeuge für PowerShell	2.x	Ende des Supports	8.11.2013	
Werkzeuge für PowerShell	3.x	Ende des Supports	29.7.2015	
Werkzeuge für PowerShell	4.x	Allgemeine Verfügbarkeit	21.11.2019	

Suchen Sie nach einem SDK Tool, das nicht erwähnt wurde? VerschlüsselungsSDKs, IoT-Geräte SDKs und Mobilgeräte SDKs sind beispielsweise nicht in diesem Handbuch enthalten. Dokumentation zu diesen anderen Tools finden Sie unter [Tools, auf denen Sie aufbauen können AWS](#).

Dokumenthistorie für AWS SDKs und Referenzhandbuch für Tools

In der folgenden Tabelle werden wichtige Ergänzungen und Aktualisierungen des AWS SDKs und Referenzhandbuch für Tools. Wenn Sie über Aktualisierungen dieser Dokumentation informiert werden möchten, können Sie den RSS Feed abonnieren.

Änderung	Beschreibung	Datum
Swift SDK zur Einstellungsreferenz hinzufügen	SDKSwift-Unterstützung zu allen Einstellungsreferenzen hinzufügen Kompatibilität mit AWS SDKs Tabellen.	17. September 2024
SDK für Java 1.x-Systemeigenschaften	Fügen Sie Details zu den unterstützten JVM Systemkonfigurationseinstellungen hinzu, indem Sie AWS SDK for Java 1.x.	30. Mai 2024
Aktualisierungen der Einstellungen	Fügen Sie JVM Systemkonfigurationseinstellungen hinzu.	27. März 2024
Aktualisierungen der Kompatibilitätstabelle	Aktualisierungen der Kompatibilität für den SDK Support, Aktualisierungen der IAM Identity Center-Verfahren.	20. Februar 2024
Aktualisierung der Container-Anmeldeinformationen. IMDS aktualisieren.	Unterstützung für Amazon hinzugefügt EKS. Einstellung zum Deaktivieren 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 die Funktionen der Tools wurden aktualisiert und umfassen SDK nun auch Kotlin, SDK Rust und AWS Tools for PowerShell.	10. Dezember 2023
Aktualisierungen der Authentifizierung	Aktualisierungen der unterstützten Authentifizierungsmethoden SDKs und Tools.	1. Juli 2023
IAMUpdates zu bewährten Verfahren	Aktualisierter Leitfaden zur Anpassung an die IAM bewährten Verfahren. Weitere Informationen finden Sie unter Bewährte Sicherheitsmethoden unter IAM .	27. Februar 2023
SSOAktualisierungen	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 über den IMDS Client und die IMDS Anmeldedaten Aktualisierungen der Umgebungsvariablen.	04. November 2022
Die Willkommenseite wird aktualisiert	Ankündigung von Amazon CodeWhisperer.	22. September 2022

Änderung des Dienstnamens für Single Sign-On	Aktualisierungen, um dies widerzuspiegeln AWS SSO wird jetzt bezeichnet als AWS IAM Identity Center.	26. Juli 2022
Aktualisierung der Einstellungen	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

AWS-Glossar

Die neueste AWS-Terminologie finden Sie im [AWS-Glossar](#) in der AWS-Glossar-Referenz.

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