



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 Marken, die nicht im Besitz von Amazon sind, gehören den jeweiligen Besitzern, die möglicherweise mit Amazon verbunden sind oder von Amazon gesponsert werden.

---

# Table of Contents

AWS Referenzhandbuch für SDKs 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 .....	11
Ändern Sie den Standardspeicherort dieser Dateien .....	12
Umgebungsvariablen .....	13
Festlegen von Umgebungsvariablen .....	14
Einrichtung von serverlosen Umgebungsvariablen .....	15
JVM-Systemeigenschaften .....	16
Wie legt man die JVM-Systemeigenschaften fest .....	16
Authentifizierung und Zugriff .....	18
AWS Builder ID .....	20
Authentifizierung von IAM Identity Center .....	20
Konfigurieren Sie den programmatischen Zugriff mithilfe von IAM Identity Center .....	21
Verstehen Sie die IAM Identity Center-Authentifizierung .....	24
IAM Roles Anywhere .....	29
Schritt 1: Konfigurieren Sie IAM Roles Anywhere .....	29
Schritt 2: Verwenden Sie IAM Roles Anywhere .....	29
Übernehmen einer Rolle .....	31
Nehmen Sie eine IAM-Rolle an .....	31
Verbunden mit Web-Identität oder OpenID Connect .....	33
AWS -Zugriffsschlüssel .....	34
Verwenden kurzfristiger Anmeldeinformationen .....	34
Verwenden langfristiger Anmeldeinformationen .....	35
Kurzfristige Anmeldeinformationen .....	36
Langfristige Anmeldeinformationen .....	38
IAM-Rollen für Amazon EC2 EC2-Instances .....	41
Erstellen einer IAM-Rolle .....	42

Starten Sie eine Amazon EC2 EC2-Instance und geben Sie Ihre IAM-Rolle an .....	42
Connect zur EC2-Instance her .....	43
Führen Sie die Beispielanwendung auf der EC2-Instance aus .....	43
Referenz zu Einstellungen .....	44
Serviceclients erstellen .....	44
Vorrang der Einstellungen .....	44
ConfigListe der Dateieinstellungen .....	46
CredentialsListe der Dateieinstellungen .....	49
Liste der Umgebungsvariablen .....	50
Liste der JVM-Systemeigenschaften .....	54
Standardisierte Anbieter von Anmeldeinformationen .....	57
Kette von Anbietern von Anmeldeinformationen .....	58
AWS Zugriffstasten .....	59
Nehmen Sie die Rolle des Anbieters an .....	62
Container-Anbieter .....	69
IAM Identity Center-Anbieter .....	72
IMDS-Anbieter .....	79
Prozessanbieter .....	84
Standardisierte Funktionen .....	88
Application ID .....	89
Amazon EC2-Instance-Metadaten .....	91
Amazon S3 Access Points .....	93
Multiregionale Amazon-S3-Zugriffspunkte .....	96
AWS-Region .....	98
AWS STS Regionalisierte Endpunkte .....	101
Dual-Stack- und FIPS-Endpunkte .....	103
Endpunkterkennung .....	106
Allgemeine Konfiguration .....	108
IMDS-Kunde .....	111
Wiederholungsverhalten .....	114
Komprimierung anfordern .....	120
Servicespezifische Endpunkte .....	123
Standardeinstellungen für intelligente Konfigurationen .....	169
Allgemeine Runtime .....	175
CRT-Abhängigkeiten .....	176
Wartungsrichtlinie .....	177

---

Übersicht .....	177
Versionsverwaltung .....	177
Lebenszyklus der SDK-Hauptversionen .....	177
Lebenszyklus von Abhängigkeiten .....	178
Methoden der Kommunikation .....	179
Matrix zur Versionsunterstützung .....	181
Dokumentverlauf .....	184
AWS-Glossar .....	187
.....	clxxxviii

# AWS Referenzhandbuch für SDKs und Tools

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

Dieses Handbuch enthält Informationen zu:

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

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

Wenn Sie Folgendes verwenden:	Die für Sie relevanten Abschnitte dieses Handbuchs sind:
<ul style="list-style-type: none"> <li>• Irgendein SDK oder Tool</li> </ul>	<a href="#">AWS Wartungsrichtlinie für SDKs und Tools</a>
<ul style="list-style-type: none"> <li>• <a href="#">AWS Cloud Development Kit (AWS CDK) Entwicklerhandbuch</a></li> <li>• <a href="#">AWS Serverless Application Model Entwickle rhandbuch</a></li> <li>• <a href="#">AWS Toolkit for Eclipse Benutzerhandbuch</a></li> <li>• <a href="#">AWS Toolkit for JetBrains Benutzerhandbuch</a></li> </ul>	<a href="#">Konfiguration</a> <a href="#">Authentifizierung und Zugriff</a> <a href="#">AWS Wartungsrichtlinie für SDKs und Tools</a>

Wenn Sie Folgendes verwenden:	Die für Sie relevanten Abschnitte dieses Handbuchs sind:
<ul style="list-style-type: none"> <li>• <a href="#">AWS Toolkit for Visual Studio Benutzerhandbuch</a></li> <li>• <a href="#">AWS Toolkit for Visual Studio Code Benutzerhandbuch</a></li> </ul>	
<ul style="list-style-type: none"> <li>• <a href="#">AWS Command Line Interface Benutzerhandbuch</a></li> <li>• <a href="#">AWS SDK for C++ Entwicklerhandbuch</a></li> <li>• <a href="#">AWS SDK for Go Entwicklerhandbuch</a></li> <li>• <a href="#">AWS SDK for Java Entwicklerhandbuch</a></li> <li>• <a href="#">AWS SDK for JavaScript Entwicklerhandbuch</a></li> <li>• <a href="#">AWS SDK for Kotlin</a></li> <li>• <a href="#">AWS SDK for .NET Entwicklerhandbuch</a></li> <li>• <a href="#">AWS SDK for PHP Entwicklerhandbuch</a></li> <li>• <a href="#">AWS SDK for Python (Boto3) Erste Schritte</a></li> <li>• <a href="#">AWS SDK for Ruby Entwicklerhandbuch</a></li> <li>• <a href="#">AWS SDK for Rust</a></li> <li>• <a href="#">AWS SDK for Swift</a></li> <li>• <a href="#">AWS Tools for Windows PowerShell Benutzerhandbuch</a></li> </ul>	<ul style="list-style-type: none"> <li>• <a href="#">Konfiguration</a></li> <li>• <a href="#">Authentifizierung und Zugriff</a></li> <li>• <a href="#">Referenz zu Einstellungen</a></li> <li>• <a href="#">AWSCommon Runtime (CRT) -Bibliotheken</a></li> <li>• <a href="#">AWS Wartungsrichtlinie für SDKs und Tools</a></li> <li>• <a href="#">AWS Versionsunterstützungsmatrix für SDKs und Tools</a></li> </ul>

## Ressourcen für Entwickler

Einen Überblick über Tools, die Ihnen bei der Entwicklung von Anwendungen helfen können, finden Sie unter [Tools AWS, auf denen Sie aufbauen](#) können AWS. Informationen zum Support finden Sie im [AWS Knowledge Center](#).

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 IDE-Toolkits (Integrated Development Environment) sind Plugins und Erweiterungen, die den Zugriff auf AWS Dienste von Ihrer IDE aus ermöglichen. Ausführliche Informationen zu den einzelnen IDE-Toolkits finden Sie in den Toolkit-Benutzerhandbüchern in der vorherigen Tabelle.

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

Weitere Informationen zu den Telemetriedaten, die in allen AWS IDE-Toolkits gesammelt wurden, finden Sie im Dokument [commonDefinitions.json im Github-Repository](#). `aws-toolkit-common`

Ausführliche Informationen zu den von den einzelnen AWS IDE-Toolkits gesammelten Telemetriedaten finden Sie in den Ressourcendokumenten in den Github-Repositorys der folgenden Toolkits: AWS

- [AWS Toolkit for Visual Studio](#)
- [AWS Toolkit for Visual Studio Code](#)
- [AWS Toolkit for JetBrains](#)

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 AWS Command Line Interface (AWS CLI), den AWS SDKs und anderen Tools verwendet wird. Konfigurationswerte werden an ein Profil angehängt, um einen Teil des SDK/Tools zu konfigurieren, wenn dieses Profil verwendet wird. Diese Dateien werden „gemeinsam genutzt“, da die Werte für alle Anwendungen, Prozesse oder SDKs in der lokalen Umgebung eines Benutzers wirksam werden.

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

### Profile

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

Das `[default]` Profil enthält die Werte, die von einem SDK- oder Tool-Vorgang verwendet werden, wenn kein bestimmtes benanntes Profil angegeben ist. Sie können auch separate Profile erstellen,

auf die Sie explizit anhand ihres Namens verweisen können. Jedes Profil kann je nach Anwendung und Szenario unterschiedliche Einstellungen und Werte verwenden.

### Note

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

## Legen Sie ein benanntes Profil fest

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

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

### **AWS\_PROFILE**- Umgebungsvariable

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

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

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

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

```
setx AWS_PROFILE "my_default_profile_name"
```

### **aws.profile**- JVM-Systemeigenschaft

Für das SDK für Kotlin auf der JVM und das SDK for Java 2.x können Sie [die aws.profile Systemeigenschaft festlegen](#). Wenn das SDK einen Dienstclient erstellt, verwendet es die

Einstellungen im genannten Profil, sofern die Einstellung nicht im Code überschrieben wird. Das SDK for Java 1.x unterstützt diese Systemeigenschaft nicht.

## 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 [Authentifizierung von IAM Identity Center](#). 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 `my-sso` 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 und Tools verfügbar, die diese Funktion enthalten. Informationen darüber, ob diese Funktion für Ihr SDK verfügbar ist, finden Sie unter [Kompatibilität mit AWS SDKs](#) Servicespezifische Endpunkte.

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

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

Das folgende Beispiel verwendet eine `services` Definition, um den Endpunkt so zu konfigurieren, dass er nur für Anfragen verwendet wird, die an den Amazon DynamoDB Dienst gestellt werden. Der `"local-dynamodb"` `services` Abschnitt wird im `profile` Abschnitt unter Verwendung des `services` Schlüssels namentlich referenziert. Der AWS-Service Identifikationsschlüssel lautet `dynamodb`. Der Unterabschnitt Amazon DynamoDB Service beginnt in der Zeile `dynamodb =`



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

## Speicherort der geteilten `credentials` Dateien `config` und Dateien

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

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

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

Betriebssystem	Standardspeicherort und Name der Dateien
Unter Linux und macOS	<code>~/.aws/config</code> <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 SDKs unterstützt werden, 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 Dateien geladen wurden. `credentials`

- 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 wJalrXUtnFEMI/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:
\> $Env:AWS_SESSION_TOKEN="AQoEXAMPLEH4aoAH0gNCAPy...truncated...zrkuWJ0gQs8IZZaIv2BXIa2R401
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 LambdaEntwicklerhandbuch.

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

- [Authentifizierung von IAM Identity Center](#)— 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.
- [Übernehmen einer Rolle](#)— 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

- [Verwenden von IAM-Rollen für Amazon EC2 EC2-Instances](#)— Verwenden Sie IAM-Rollen, um Ihre Anwendung sicher auf einer Amazon EC2 EC2-Instance auszuführen.
- Sie können auf folgende Weise programmgesteuert mit der AWS Nutzung von IAM Identity Center interagieren:
  - Wird verwendet [AWS CloudShell](#), um AWS CLI Befehle von der Konsole aus auszuführen.
  - Wird verwendet [AWS Cloud9](#), um AWS mit der Programmierung unter Verwendung einer integrierten Entwicklungsumgebung (IDE) mit AWS Ressourcen zu beginnen.
  - 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, 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 dann dieses Token in AWS gegen temporäre Sicherheitsanmeldeinformationen eintauschen, die einer IAM-Rolle mit Berechtigungen zur Verwendung der Ressourcen in Ihrem AWS-Konto zugeordnet sind.

Wie Sie dies für Ihr SDK oder Tool konfigurieren, erfahren Sie unter [Verbunden mit Web-Identität oder OpenID Connect](#)

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-Referenzhandbuch 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 CodeWhisperer 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.
- [Authentifizierung mit CodeWhisperer und AWS Toolkit — Builder ID](#) im CodeWhisperer Benutzerhandbuch — Erfahren Sie, wie Sie eine CodeWhisperer AWS Builder ID verwenden.
- [CodeCatalystKonzepte — AWS Builder ID](#) im CodeCatalyst Amazon-Benutzerhandbuch — Erfahren Sie, wie ein CodeCatalyst verwendet wird AWS Builder ID.

## Authentifizierung von IAM Identity Center

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 (Amazon EC2) oder entwickeln, empfehlen wir AWS Cloud9, stattdessen Anmeldeinformationen von diesem Service zu beziehen.

In diesem Tutorial richten Sie den Zugriff auf das IAM Identity Center 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 der 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 Zugriffportal, um Konfigurationswerte abzurufen, die Sie für den SDK/Tool-Authentifizierungsprozess benötigen.
- Das AWS CLI wird verwendet, um Ihr SDK oder Tool so zu konfigurieren, dass es die IAM Identity Center-Authentifizierung für API-Aufrufe verwendet, die über Ihren Code getätigt werden. Dieser einmalige Vorgang aktualisiert Ihre gemeinsam genutzte AWS `config` Datei, die dann von Ihrem SDK oder Tool verwendet wird, wenn Sie Ihren Code ausführen.

# 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 [Aktivieren von IAM Identity Center](#) 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 diese Option PowerUserAccess, wenn Sie mit Ihrem Tool/SDK arbeiten.

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

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

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

Ich habe bereits AWS über das von meinem Arbeitgeber verwaltete AWS Zugriffsportal Zugriff darauf

Melden Sie sich AWS über das AWS Zugangsportal 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.

Ich habe bereits AWS über einen föderierten Anbieter für benutzerdefinierte Identitäten, der von meinem Arbeitgeber verwaltet wird, Zugriff darauf

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

## Schritt 2: Konfigurieren Sie SDKs und Tools für die Verwendung von IAM Identity Center

1. Installieren Sie auf Ihrem Entwicklungscomputer die neueste Version. AWS CLI
  - a. Weitere Informationen finden Sie [im AWS Command Line Interface Benutzerhandbuch unter Installation oder Aktualisierung AWS CLI der neuesten Version von.](#)
  - b. (Optional) Um zu überprüfen, ob der AWS CLI funktioniert, öffnen Sie eine Befehlszeile und führen Sie den `aws --version` Befehl aus.

2. Melden Sie sich beim AWS Access Portal an. Ihr Arbeitgeber kann diese URL angeben oder Sie erhalten sie in einer E-Mail nach Schritt 1: Zugriff einrichten. Wenn nicht, finden Sie die URL Ihres AWS Zugangsportals im Dashboard von <https://console.aws.amazon.com/singlesignon/>.
  - a. Wählen Sie im AWS Zugriffportal 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 `SSO Start URL` und Werte zu erhalten, die Sie für den nächsten Schritt benötigen.
3. Führen AWS CLI Sie den Befehl in der `aws configure sso` Befehlszeile aus. 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](#).
  - Für den CLI-Profilnamen empfehlen wir, zu Beginn die *StandardEinstellung* einzugeben. Informationen darüber, wie Sie nicht standardmäßige (benannte) Profile und die zugehörige Umgebungsvariable einrichten, 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 von Ihnen konfigurierte IAM Identity Center-Berechtigungssatz angezeigt werden.
5. Wenn Sie ein AWS SDK verwenden, erstellen Sie eine Anwendung für Ihr SDK in Ihrer Entwicklungsumgebung.
  - a. Bei einigen SDKs `SSOIDC` müssen zusätzliche Pakete wie `SSO` und zu Ihrer Anwendung hinzugefügt werden, bevor Sie die IAM Identity Center-Authentifizierung verwenden können. Einzelheiten finden Sie in Ihrem spezifischen SDK.
  - b. Wenn Sie den Zugriff auf zuvor konfiguriert haben AWS, überprüfen Sie Ihre geteilte `AWS credentials` Datei auf etwaige [AWS Zugriffstasten](#). Aufgrund der [Kette von Anbietern von Anmeldeinformationen](#) Rangfolge müssen Sie alle statischen Anmeldeinformationen

entfernen, bevor das SDK oder das Tool die IAM Identity Center-Anmeldeinformationen verwendet.

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

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

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

Einzelheiten zu allen IAM Identity Center-Anbiereinstellungen für 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

Name des IAM Identity Center	SDK-API-Name	Beschreibung
		die sso API-Namespace aus Gründen der Abwärtskompatibilität ihren ursprünglichen Namen. Weitere Informationen finden Sie unter Umbenennung von <a href="#">IAM Identity Center im Benutzerhandbuch</a> . 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.
Sitzung mit Berechtigungssatz		Die IAM-Sitzung, die das SDK intern für die AWS-Service Aufrufe verwendet. In informellen Diskussionen wird dies möglicherweise fälschlicherweise als „Rollensitzung“ bezeichnet.

Name des IAM Identity Center	SDK-API-Name	Beschreibung
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 Aktualisierungs- und Zugriffstoken. AWS CLIAußerdem 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 CenterBenutzer](#).
- 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 instanzieren. 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 CenterBenutzerhandbuch.
- Beachten Sie, dass die Berechtigungssatz-Anmeldeinformationen in den meisten AWS SDK-API-Dokumentationen auch als AWSAnmeldeinformationen 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 tätigenAWS-Service, z. B. Amazon S3 aufzufordern,



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

### Ablauf und Aktualisierung der Sitzung

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

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

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

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

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

# 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 Prozessanmeldeinformationen](#) in diesem Handbuch.

## Übernehmen einer Rolle

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 ServiceAPI-Referenz unter [Aktionen](#).

Um Ihr SDK oder Tool so einzurichten, dass es eine Rolle übernimmt, müssen Sie zunächst eine bestimmte Rolle erstellen oder identifizieren, die Sie übernehmen möchten. IAM-Rollen werden durch eine Rolle mit dem Amazon Resource Name ([ARN](#)) eindeutig identifiziert. Rollen bauen Vertrauensbeziehungen zu einer anderen Entität auf. Die vertrauenswürdige Entität, die die Rolle verwendet, kann ein anderes AWS-ServiceAWS-Konto, ein Web-Identitätsanbieter oder ein OIDC- oder SAML-Verbund sein. Weitere Informationen zu IAM-Rollen finden Sie unter [Verwenden von IAM-Rollen im IAM-Benutzerhandbuch](#).

Nachdem die IAM-Rolle identifiziert wurde und Sie diese Rolle als vertrauenswürdig eingestuft haben, können Sie Ihr SDK oder Tool so konfigurieren, dass die von der Rolle gewährten Berechtigungen verwendet werden. Um dies zu tun, entweder [Nehmen Sie eine IAM-Rolle an](#) oder [Verbunden mit Web-Identität oder OpenID Connect](#).

### Nehmen Sie eine IAM-Rolle an

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. Andere Beispiele für die Übernahme einer Rolle sind die Verwaltung mehrerer Rollen AWS-Konten von Amazon EC2 aus, die Nutzung von AWS CodeCommit Across AWS-Konten oder der Zugriff auf ein anderes Konto vonAWS CodeBuild.

#### Schritt 1: Richten Sie eine IAM-Rolle ein

Um Ihr SDK oder Tool so einzurichten, dass es eine Rolle übernimmt, müssen Sie zunächst eine bestimmte Rolle erstellen oder identifizieren, die Sie übernehmen möchten. IAM-Rollen werden mithilfe eines [Rollen-ARN](#) eindeutig identifiziert. Rollen stellen Vertrauensbeziehungen zu einer

anderen Entität her, in der Regel innerhalb Ihres Kontos oder für kontoübergreifenden Zugriff. Informationen zur Einrichtung finden Sie unter [Erstellen von IAM-Rollen](#) im IAM-Benutzerhandbuch.

## Schritt 2: Konfigurieren Sie das SDK oder das Tool

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

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

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

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

- `role_arn`— Aus der IAM-Rolle, die Sie in Schritt 1 erstellt haben
- Konfigurieren Sie entweder `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.

## Verbunden mit Web-Identität oder OpenID Connect

Gibt beim Erstellen von mobilen Anwendungen oder clientbasierten Webanwendungen, für die Zugriff erforderlich ist AWS, einen Satz temporärer Sicherheitsanmeldedaten für Verbundbenutzer AWS STS zurück, die über einen Public Identity Provider (IdP) authentifiziert wurden. Beispiele für öffentliche Identitätsanbieter sind Login with Amazon, Facebook, Google oder ein beliebiger OpenID Connect-kompatibler Anbieter (OIDC). Bei dieser Methode benötigen Ihre Benutzer keine eigenen AWS Identitäten oder IAM-Identitäten.

Wenn Sie Amazon Elastic Kubernetes Service verwenden, bietet diese Funktion die Möglichkeit, unterschiedliche IAM-Rollen für jeden Ihrer Container anzugeben. Kubernetes bietet die Möglichkeit, OIDC-Token an Ihre Container zu verteilen, die von diesem Anmeldeinformationsanbieter zum Abrufen temporärer Anmeldeinformationen verwendet werden. Weitere Informationen zu dieser Amazon EKS-Konfiguration finden Sie unter [IAM-Rollen für Dienstkonten](#) im Amazon EKS-Benutzerhandbuch. Für eine einfachere Option empfehlen wir jedoch, stattdessen [Amazon EKS Pod Identities](#) zu verwenden, sofern Ihr [SDK dies unterstützt](#).

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

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

### Schritt 2: Konfigurieren Sie das SDK oder das Tool

Konfigurieren Sie das SDK oder das Tool so, dass es ein Web-Identitätstoken von AWS STS für die Authentifizierung verwendet.

Wenn Sie dies in einem Profil angeben, führt das SDK oder Tool automatisch den entsprechenden AWS STS [AssumeRoleWithWebIdentity](#) API-Aufruf für Sie durch. Um temporäre

Anmeldeinformationen mithilfe des Web Identity Federation abzurufen und zu verwenden, geben Sie die folgenden Konfigurationswerte in der gemeinsam genutzten AWS config Datei an. Weitere Informationen zu den einzelnen Einstellungen finden Sie im [Nehmen Sie die Einstellungen des Anbieters für Anmeldeinformationen an](#) Abschnitt.

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

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

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

#### Note

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

Einzelheiten zu allen Einstellungen des Anbieters von Anmeldeinformationen 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 [Authentifizierung von IAM Identity Center](#) 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 oder Tool so zu konfigurieren, dass es [Authentifizierung von IAM Identity Center](#) mit Optionen für die erweiterte Sitzungsdauer verwendet wird. Sie können jedoch temporäre Anmeldeinformationen, die im AWS-Zugriffportal 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.

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](#).



## Authentifizieren Sie sich mit langfristigen 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](#).

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

## Wichtige Warnhinweise und Richtlinien für Anmeldeinformationen

### Warnhinweise für Anmeldeinformationen

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

### Zusätzliche Hinweise zur sicheren Verwaltung von Anmeldeinformationen

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

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

## Voraussetzungen: Erstellen Sie ein AWS Konto

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

1. Erstellen Sie ein Konto.

Informationen zum Erstellen eines AWS Kontos finden Sie unter [Erste Schritte: Sind Sie ein Erstbenutzer? AWS](#) im AWS 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 unter [Erstellen eines Administratorbenutzers](#) im IAM-Benutzerhandbuch beschrieben.

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

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

## Schritt 1: Erstellen Ihres IAM-Benutzers

- Erstellen Sie Ihren IAM-Benutzer, indem Sie das Verfahren [Erstellen von IAM-Benutzern \(Konsole\)](#) im IAM-Benutzerhandbuch befolgen. Gehen Sie beim Erstellen Ihres IAM-Benutzers wie folgt vor:
  - Wir empfehlen Ihnen, Benutzerzugriff auf die AWS Management Console bereitzustellen auszuwählen. Auf diese Weise können Sie den Code, den Sie gerade ausführen, in einer visuellen Umgebung anzeigen AWS-Services , z. B. beim Überprüfen von AWS CloudTrail

Diagnoseprotokollen oder beim Hochladen von Dateien in Amazon Simple Storage Service, was beim Debuggen Ihres Codes hilfreich ist.

- Wählen Sie unter Berechtigungen festlegen — Berechtigungsoptionen die Option Richtlinien direkt anhängen aus, um festzulegen, wie Sie diesem Benutzer Berechtigungen zuweisen möchten.
- Die meisten SDK-Tutorials zum Thema „Erste Schritte“ verwenden den Amazon-S3-Service als Beispiel. Wenn Sie Ihrer Anwendung Vollzugriff auf Amazon S3 gewähren möchten, wählen Sie die AmazonS3FullAccess-Richtlinie zum Anfügen an diesen Benutzer aus.
- Sie können die optionalen Schritte dieses Verfahrens zum Festlegen von Berechtigungsgrenzen oder Tags ignorieren.

## Schritt 2: Abrufen Ihrer Zugriffsschlüssel

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

### Warning

Verwenden Sie geeignete Sicherheitsmaßnahmen, um diese Anmeldeinformationen zu schützen.

6. Wählen Sie Done (Fertig).

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

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

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

3. Speichern Sie die Datei.

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

## Verwenden von IAM-Rollen für Amazon EC2 EC2-Instances

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

Erstellen Sie für eine Amazon Elastic Compute Cloud-Instance eine IAM-Rolle und gewähren Sie dann Ihrer Amazon EC2 EC2-Instance Zugriff auf diese Rolle. Weitere Informationen finden Sie unter [IAM-Rollen für Amazon EC2](#) im Amazon EC2-Benutzerhandbuch oder [IAM-Rollen für Amazon EC2 im Amazon EC2 EC2-Benutzerhandbuch](#).

## Erstellen einer IAM-Rolle

Erstellen Sie eine IAM-Rolle, die nur Lesezugriff auf Amazon S3 gewährt.

1. [Melden Sie sich bei der an AWS Management Console und öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.](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 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 Amazon EC2 EC2-Instance starten.

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

Sie können eine Amazon EC2 EC2-Instance mit einer IAM-Rolle über die Amazon EC2 EC2-Konsole starten.

Folgen Sie den Anweisungen zum Starten einer Instance im [Amazon EC2 EC2-Benutzerhandbuch](#) oder im [Amazon EC2 EC2-Benutzerhandbuch](#).

Wenn Sie die Seite Review Instance Launch erreichen, klicken Sie auf Edit instance details. Wählen Sie unter IAM-Rolle die IAM-Rolle aus, die Sie zuvor erstellt haben. Befolgen Sie die Anweisungen und schließen Sie den Vorgang ab.

### Note

Zum Herstellen einer Verbindung mit der Instance müssen Sie eine Sicherheitsgruppe und ein Schlüsselpaar neu erstellen oder vorhandene Anmeldeinformationen auswählen.

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

## Connect zur EC2-Instance her

Stellen Sie eine Verbindung mit der EC2-Instance her, sodass Sie die Beispielanwendung auf diese übertragen und dann die Anwendung ausführen können. Sie benötigen die Datei, die den privaten Teil des key pair enthält, das Sie zum Starten der Instance verwendet haben, d. h. die PEM-Datei.

Sie können dies tun, indem Sie das Verbindungsverfahren im [Amazon EC2 EC2-Benutzerhandbuch](#) oder im [Amazon EC2 EC2-Benutzerhandbuch befolgen](#). Wenn Sie eine Verbindung herstellen, tun Sie dies so, dass Sie Dateien von Ihrem Entwicklungscomputer auf Ihre Instance übertragen können.

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

## Führen Sie die Beispielanwendung auf der EC2-Instance aus

1. Kopieren Sie die Anwendungsdateien von Ihrem lokalen Laufwerk auf Ihre Instance.

Informationen zum Übertragen von Dateien auf Ihre Instance finden Sie im [Amazon EC2 EC2-Benutzerhandbuch](#) oder im [Amazon EC2 EC2-Benutzerhandbuch](#).

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 Amazon EC2 EC2-Konsole an AWS Management Console und öffnen Sie sie unter <https://console.aws.amazon.com/ec2/>.
  - b. Wählen Sie die Instance aus und trennen Sie die IAM-Rolle über Aktionen, Instance-Einstellungen, IAM-Rolle anfügen/ersetzen.
  - c. Führen Sie die Anwendung erneut aus und vergewissern Sie sich, dass sie einen Autorisierungsfehler zurückgibt.



# Referenz zu Einstellungen

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

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

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

## Serviceclients erstellen

Für den programmgesteuerten Zugriff AWS-Services verwenden SDKs jeweils eine Clientklasse/ein Client-Objekt. AWS-Service Wenn Ihre Anwendung beispielsweise auf Amazon EC2 zugreifen muss, erstellt Ihre Anwendung ein Amazon EC2 EC2-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 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 SDKs unterstützt werden und weitreichende Auswirkungen auf alle haben. AWS-Services Alle SDKs haben eine Reihe von Orten (oder Quellen), die überprüft werden, 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 und Tools wird die Prüfung möglicherweise in einer anderen Reihenfolge durchgeführt. Einige SDKs und 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.

## 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
api_versions	<a href="#">Allgemeine Konfigurationseinstellungen</a>
aws_access_key_id	<a href="#">AWS Zugriffstasten</a>
aws_secret_access_key	<a href="#">AWS Zugriffstasten</a>
aws_session_token	<a href="#">AWS Zugriffstasten</a>
ca_bundle	<a href="#">Allgemeine Konfigurationseinstellungen</a>
credential_process	<a href="#">Anbieter für Prozessanmeldeinformationen</a>
credential_source	<a href="#">Nehmen Sie die Rolle des Anbieters von Anmeldeinformationen an</a>
defaults_mode	<a href="#">Standardeinstellungen für intelligente Konfigurationen</a>
disable_request_compression	<a href="#">Komprimierung anfordern</a>

Einstellungsname	Details
duration_seconds	<a href="#">Nehmen Sie die Rolle des Anbieters von Anmeldeinformationen an</a>
ec2_metadata_service_endpoint	<a href="#">Anbieter von IMDS-Anmeldeinformationen</a>
ec2_metadata_service_endpoint_mode	<a href="#">Anbieter von IMDS-Anmeldeinformationen</a>
ec2_metadata_v1_disabled	<a href="#">Anbieter von IMDS-Anmeldeinformationen</a>
endpoint_discovery_enabled	<a href="#">Erkennung von Endpunkten</a>
endpoint_url	<a href="#">Servicespezifische Endpunkte</a>
external_id	<a href="#">Nehmen Sie die Rolle des Anbieters von Anmeldeinformationen an</a>
ignore_configured_endpoint_urls	<a href="#">Dienstspezifische Endpunkte</a>
max_attempts	<a href="#">Verhalten wiederholen</a>
metadata_service_number_attempts	<a href="#">Amazon EC2 EC2-Instance-Metadaten</a>

Einstellungsname	Details
metadata_service_timeout	<a href="#">Amazon EC2 EC2-Instance-Metadaten</a>
mfa_serial	<a href="#">Übernehmen Sie die Rolle des Anbieters von Anmeldeinformationen</a>
output	<a href="#">Allgemeine Konfigurationseinstellungen</a>
parameter_validation	<a href="#">Allgemeine Konfigurationseinstellungen</a>
region	<a href="#">AWS-Region</a>
request_min_compression_size_bytes	<a href="#">Komprimierung anfordern</a>
retry_mode	<a href="#">Verhalten erneut versuchen</a>
role_arn	<a href="#">Nehmen Sie die Rolle des Anbieters von Anmeldeinformationen an</a>
role_session_name	<a href="#">Nehmen Sie die Rolle des Anbieters von Anmeldeinformationen an</a>
s3_disable_multiregion_access_points	<a href="#">Multiregionale Amazon-S3-Zugriffspunkte</a>
s3_use_arn_region	<a href="#">Amazon-S3-Zugriffspunkte</a>
sdk_ua_app_id	<a href="#">Application ID</a>

Einstellungsname	Details
source_profile	<a href="#">Nehmen Sie die Rolle des Anbieters von Anmeldeinformationen an</a>
sso_account_id	<a href="#">Anbieter von IAM Identity Center-Anmeldeinformationen</a>
sso_region	<a href="#">Anbieter von IAM Identity Center-Anmeldeinformationen</a>
sso_registration_scopes	<a href="#">Anbieter von IAM Identity Center-Anmeldeinformationen</a>
sso_role_name	<a href="#">Anbieter von IAM Identity Center-Anmeldeinformationen</a>
sso_start_url	<a href="#">Anbieter von IAM Identity Center-Anmeldeinformationen</a>
sts_regional_endpoints	<a href="#">AWS STS Regionalisierte Endpunkte</a>
use_dualstack_endpoint	<a href="#">Dual-Stack- und FIPS-Endpunkte</a>
use_fips_endpoint	<a href="#">Dual-Stack- und FIPS-Endpunkte</a>
web_identity_token_file	<a href="#">Nehmen Sie die Rolle des Anbieters von Anmeldeinformationen an</a>

## 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
aws_access_key_id	<a href="#">AWS Zugriffstasten</a>
aws_secret_access_key	<a href="#">AWS Zugriffstasten</a>
aws_session_token	<a href="#">AWS Zugriffstasten</a>

## Liste der Umgebungsvariablen

Die von den meisten SDKs unterstützten Umgebungsvariablen 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
AWS_ACCESS_KEY_ID	<a href="#">AWS Zugriffstasten</a>
AWS_CA_BUNDLE	<a href="#">Allgemeine Konfigurationseinstellungen</a>
AWS_CONFIG_FILE	<a href="#">Speicherort der geteilten credentials Dateien config und Dateien</a>
AWS_CONTAINER_AUTHORIZATION_TOKEN	<a href="#">Anbieter von Container-Anmeldeinformationen</a>

Einstellungsname	Details	
AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE	<a href="#">Anbieter von Container-Anmeldeinformationen</a>	
AWS_CONTAINER_CREDENTIALS_FULL_URI	<a href="#">Anbieter von Container-Anmeldeinformationen</a>	
AWS_CONTAINER_CREDENTIALS_RELATIVE_URI	<a href="#">Anbieter von Container-Anmeldeinformationen</a>	
AWS_DEFAULTS_MODE	<a href="#">Standardeinstellungen für intelligente Konfigurationen</a>	
AWS_DISABLE_REQUEST_COMPRESSION	<a href="#">Komprimierung anfordern</a>	
AWS_EC2_METADATA_DISABLED	<a href="#">Anbieter von IMDS-Anmeldeinformationen</a>	
AWS_EC2_METADATA_SERVICE_ENDPOINT	<a href="#">Anbieter von IMDS-Anmeldeinformationen</a>	
AWS_EC2_METADATA_SERVICE_ENDPOINT_MODE	<a href="#">Anbieter von IMDS-Anmeldeinformationen</a>	



Einstellungsname	Details
AWS_EC2_METADATA_DISABLED	<a href="#">Anbieter von IMDS-Anmeldeinformationen</a>
AWS_ENABLE_ENDPOINT_DISCOVERY	<a href="#">Erkennung von Endpunkten</a>
AWS_ENDPOINT_URL	<a href="#">Servicespezifische Endpunkte</a>
AWS_ENDPOINT_URL_<SERVICE>	<a href="#">Servicespezifische Endpunkte</a>
AWS_IAM_ROLE_ARN	<a href="#">Nehmen Sie die Rolle des Anbieters von Anmeldeinformationen an</a>
AWS_IAM_ROLE_SESSION_NAME	<a href="#">Nehmen Sie die Rolle des Anbieters von Anmeldeinformationen an</a>
AWS_IGNORE_ENDPOINT_URLS	<a href="#">Dienstspezifische Endpunkte</a>
AWS_MAX_ATTEMPTS	<a href="#">Verhalten wiederholen</a>
AWS_METADATA_SERVICE_NUM_ATTEMPTS	<a href="#">Amazon EC2 EC2-Instance-Metadaten</a>

Einstellungsname	Details
AWS_METAD ATA_SERVI CE_TIMEOUT	<a href="#">Amazon EC2 EC2-Instance-Metadaten</a>
AWS_PROFILE	<a href="#">Geteilte Dateien config und Dateien credentials</a>
AWS_REGION	<a href="#">AWS-Region</a>
AWS_REQUE ST_MIN_CO MPRESSION _SIZE_BYTES	<a href="#">Komprimierung anfordern</a>
AWS_RETRY_MODE	<a href="#">Verhalten erneut versuchen</a>
AWS_S3_DI SABLE_MUL TIREGION_ ACCESS_POINTS	<a href="#">Multiregionale Amazon-S3-Zugriffspunkte</a>
AWS_S3_US E_ARN_REGION	<a href="#">Amazon-S3-Zugriffspunkte</a>
AWS_SDK_U A_APP_ID	<a href="#">Application ID</a>
AWS_SECRE T_ACCESS_KEY	<a href="#">AWS Zugriffstasten</a>
AWS_SESSI ON_TOKEN	<a href="#">AWS Zugriffstasten</a>
AWS_SHARE D_CREDENT IALS_FILE	<a href="#">Speicherort der geteilten credentials Dateien config und Dateien</a>

Einstellungsname	Details
AWS_STS_REGIONAL_ENDPOINTS	<a href="#">AWS STS Regionalisierte Endpunkte</a>
AWS_USE_DUALSTACK_ENDPOINT	<a href="#">Dual-Stack- und FIPS-Endpunkte</a>
AWS_USE_FIPS_ENDPOINT	<a href="#">Dual-Stack- und FIPS-Endpunkte</a>
AWS_WEB_IDENTITY_TOKEN_FILE	<a href="#">Nehmen Sie die Rolle des Anbieters von Anmeldeinformationen an</a>

## 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>	<a href="#">AWS Zugriffstasten</a>
<code>aws.configFile</code>	<a href="#">Speicherort der geteilten credentials Dateien config und Dateien</a>
<code>aws.defaultMode</code>	<a href="#">Standardeinstellungen für die intelligente Konfiguration</a>
<code>aws.disableEc2MetadataV1</code>	<a href="#">Anbieter von IMDS-Anmeldeinformationen</a>

Einstellungsname	Details
<code>aws.disableRequestCompression</code>	<a href="#">Komprimierung anfordern</a>
<code>aws.ec2MetadataServiceEndpoint</code>	<a href="#">Anbieter von IMDS-Anmeldeinformationen</a>
<code>aws.ec2MetadataEndpointMode</code>	<a href="#">Anbieter von IMDS-Anmeldeinformationen</a>
<code>aws.endpointDiscoveryEnabled</code>	<a href="#">Erkennung von Endpunkten</a>
<code>aws.endpointUrl</code>	<a href="#">Servicespezifische Endpunkte</a>
<code>aws.endpointUrl&lt;ServiceName&gt;</code>	<a href="#">Servicespezifische Endpunkte</a>
<code>aws.ignoreConfiguredEndpointUrls</code>	<a href="#">Servicespezifische Endpunkte</a>
<code>aws.maxAttempts</code>	<a href="#">Verhalten wiederholen</a>
<code>aws.profile</code>	<a href="#">Geteilte Dateien config und Dateien credentials</a>
<code>aws.region</code>	<a href="#">AWS-Region</a>

Einstellungsname	Details
<code>aws.requestMinCompressionSizeBytes</code>	<a href="#">Komprimierung anfordern</a>
<code>aws.retryMode</code>	<a href="#">Verhalten erneut versuchen</a>
<code>aws.roleArn</code>	<a href="#">Nehmen Sie die Rolle des Anbieters von Anmeldeinformationen an</a>
<code>aws.roleSessionName</code>	<a href="#">Nehmen Sie die Rolle des Anbieters von Anmeldeinformationen an</a>
<code>aws.s3DisableMultiRegionAccessPoints</code>	<a href="#">Multiregionale Amazon-S3-Zugriffspunkte</a>
<code>aws.s3UseArnRegion</code>	<a href="#">Amazon-S3-Zugriffspunkte</a>
<code>aws.secretAccessKey</code>	<a href="#">AWS Zugriffstasten</a>
<code>aws.sessionToken</code>	<a href="#">AWS Zugriffstasten</a>
<code>aws.shareCredentialsFile</code>	<a href="#">Speicherort der geteilten credentials Dateien config und Dateien</a>
<code>aws.useDualstackEndpoint</code>	<a href="#">Dual-Stack- und FIPS-Endpunkte</a>
<code>aws.useFipsEndpoint</code>	<a href="#">Dual-Stack- und FIPS-Endpunkte</a>

Einstellungsname	Details
<code>aws.userAgentAppId</code>	<a href="#">Application ID</a>
<code>aws.webIdentityTokenFile</code>	<a href="#">Nehmen Sie die Rolle des Anbieters von Anmeldeinformationen an</a>

## AWS SDKs und Tools standardisierte Anbieter von Anmeldeinformationen

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

### Important

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

### Themen

- [Kette von Anbietern von Anmeldeinformationen](#)
- [AWS Zugriffstasten](#)
- [Übernehmen Sie die Rolle Credential Provider](#)
- [Anbieter von Container-Anmeldeinformationen](#)
- [IAM Identity Center-Anmeldeinformationsanbieter](#)
- [IMDS-Anmeldeinformationsanbieter](#)
- [Anbieter von Prozessanmeldeinformationen](#)

## Kette von Anbietern von Anmeldeinformationen

Alle SDKs verfügen über eine Reihe von Stellen (oder Quellen), an denen sie nach gültigen Anmeldeinformationen suchen, mit denen eine Anfrage an eine gesendet werden kann. AWS-Service Nachdem gültige Anmeldeinformationen gefunden wurden, wird die Suche beendet. Diese systematische Suche wird als Standardanbieterkette für Anmeldeinformationen bezeichnet.

Die von den einzelnen SDKs verwendete Kette ist zwar unterschiedlich, sie enthalten jedoch am häufigsten Quellen wie die folgenden:

Anbieter von Anmeldeinformationen	Beschreibung
<a href="#">AWS Zugriffstasten</a>	AWS Zugriffsschlüssel für einen IAM-Benutzer (wie <code>AWS_ACCESS_KEY_ID</code> , und <code>AWS_SECRET_ACCESS_KEY</code> ).
<a href="#">Verbunden mit Web-Identität oder OpenID Connect</a> — 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 Web-Identitätstoken von ( ) verwenden. AWS Security Token Service AWS STS
<a href="#">IAM Identity Center-Anmeldeinformationsanbieter</a>	Holen Sie sich Anmeldeinformationen von AWS IAM Identity Center.
<a href="#">Übernehmen Sie die Rolle Credential Provider</a>	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).
<a href="#">Anbieter von Container-Anmeldeinformationen</a>	Anmeldeinformationen für Amazon Elastic Container Service (Amazon ECS) und Amazon Elastic Kubernetes Service (Amazon EKS). Der Anbieter für Container-Anmeldeinformationen ruft Anmeldeinformationen für die containerisierte Anwendung des Kunden ab.
<a href="#">Anbieter von Prozessanmeldeinformationen</a>	Benutzerdefinierter Anbieter für Anmeldeinformationen. Rufen Sie Ihre Anmeldeinformationen aus einer externen

Anbieter von Anmeldeinformationen	Beschreibung
	Quelle oder einem externen Prozess ab, einschließlich IAM Roles Anywhere.
<a href="#">IMDS-Anmeldeinformationsanbieter</a>	Anmeldeinformationen für das Amazon Elastic Compute Cloud (Amazon EC2) -Instanzprofil. Ordnen Sie jeder Ihrer EC2-Instances eine IAM-Rolle zu. Temporäre Anmeldeinformationen für diese Rolle werden dem Code zur Verfügung gestellt, der in der Instance ausgeführt wird. Die Anmeldeinformationen werden über den Amazon-EC2-Metadaten-Service bereitgestellt.

Für jeden Schritt in der Kette gibt es mehrere Möglichkeiten, Einstellungswerte zuzuweisen. Einstellungswerte, die im Code angegeben sind, haben immer Vorrang. Es gibt jedoch auch [Umgebungsvariablen](#) und die [Geteilte credentials Dateien config und Dateien](#). Weitere Informationen finden Sie unter [Vorrang der Einstellungen](#).

## AWS Zugriffstasten

### Warning

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

AWS Zugriffsschlüssel für einen IAM-Benutzer können als Ihre AWS Anmeldeinformationen verwendet werden. Das AWS SDK verwendet diese AWS Anmeldeinformationen automatisch, um API-Anfragen zu signieren AWS, sodass Ihre Workloads sicher und bequem auf Ihre AWS Ressourcen und Daten zugreifen können. Es wird empfohlen, immer die zu verwenden, `aws_session_token` damit die Anmeldeinformationen temporär sind und nach Ablauf nicht mehr gültig sind. Die Verwendung langfristiger Anmeldeinformationen wird nicht empfohlen.



**Note**

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

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

Konfigurieren Sie diese Funktionalität wie folgt:

**aws\_access\_key\_id**- Einstellung für gemeinsam genutzte AWS `config` Dateien,  
**aws\_access\_key\_id**- Einstellung für gemeinsam genutzte AWS `credentials` Dateien (empfohlene Methode), **AWS\_ACCESS\_KEY\_ID**- Umgebungsvariable, **aws.accessKeyId**- JVM-Systemeigenschaft: Nur Java/Kotlin

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

**aws\_secret\_access\_key**- Einstellung für gemeinsam genutzte AWS `config` Dateien,  
**aws\_secret\_access\_key**- Einstellung für gemeinsam genutzte AWS `credentials` Dateien (empfohlene Methode), **AWS\_SECRET\_ACCESS\_KEY**- Umgebungsvariable, **aws.secretAccessKey**- JVM-Systemeigenschaft: Nur Java/Kotlin

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

**aws\_session\_token**- Einstellung für gemeinsam genutzte AWS `config` Dateien,  
**aws\_session\_token**- Einstellung für gemeinsam genutzte AWS `credentials` Dateien (empfohlene Methode), **AWS\_SESSION\_TOKEN**- Umgebungsvariable, **aws.sessionToken**- JVM-Systemeigenschaft: Nur Java/Kotlin

Gibt ein AWS Sitzungstoken an, das als Teil der Anmeldeinformationen zur Authentifizierung des Benutzers verwendet wird. Sie erhalten diesen Wert als Teil der temporären Anmeldeinformationen, die bei erfolgreichen Anfragen zur Übernahme einer Rolle

zurückgegeben werden. Ein Sitzungs-Token ist nur erforderlich, wenn Sie manuell temporäre Anmeldeinformationen angeben. Wir empfehlen jedoch, immer temporäre Sicherheitsanmeldedaten statt langfristiger Anmeldeinformationen zu verwenden. Sicherheitsempfehlungen finden Sie unter [Bewährte Sicherheitsmethoden in IAM](#).

Anweisungen zum Abrufen dieser Werte finden Sie 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=wJalrXUtnFEMI/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 wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY
setx
  AWS_SESSION_TOKEN AQoEXAMPLEH4aoAH0gNCAPy...truncated...zrkuWJ0gQs8IZZaIv2BXIa2R40lgk
```

## Kompatibilität mit AWS SDKs

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

SDK	U  Hinweise oder weitere Informationen zt
<a href="#">AWS CLI v2</a>	Ja

SDK	U zt	Hinweise oder weitere Informationen
<a href="#">SDK for C++</a>	Ja	Die gemeinsam genutzte config Datei wird nicht unterstützt.
<a href="#">SDK for Go V2 (1.x)</a>	Ja	
<a href="#">SDK for Go 1.x (V1)</a>	Ja	Um die Einstellungen für gemeinsam genutzte config Dateien zu verwenden, müssen Sie das Laden aus der Konfigurationsdatei aktivieren; siehe <a href="#">Sessions</a> .
<a href="#">SDK for Java 2.x</a>	Ja	
<a href="#">SDK for Java 1.x</a>	Ja	
<a href="#">SDK für 3.x JavaScript</a>	Ja	
<a href="#">SDK für 2.x JavaScript</a>	Ja	
<a href="#">SDK für Kotlin</a>	Ja	
<a href="#">SDK for .NET 3.x</a>	Ja	Umgebungsvariablen werden nicht unterstützt.
<a href="#">SDK for PHP 3.x</a>	Ja	
<a href="#">SDK for Python (Boto3)</a>	Ja	
<a href="#">SDK for Ruby 3.x</a>	Ja	
<a href="#">SDK für Rust</a>	Ja	
<a href="#">Tools für PowerShell</a>	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 oder Tool so einzurichten, dass es eine Rolle übernimmt, müssen Sie zunächst eine bestimmte Rolle erstellen oder identifizieren, die Sie übernehmen möchten. IAM-Rollen werden durch eine Rolle mit dem Amazon Resource Name ([ARN](#)) eindeutig identifiziert. Rollen bauen Vertrauensbeziehungen zu einer anderen Entität auf. Die vertrauenswürdige Entität, die die Rolle verwendet, kann ein anderer AWS-Service AWS-Konto, ein Web-Identitätsanbieter oder ein OIDC- oder SAML-Verbund sein.

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

Anleitungen zu den ersten Schritten mit diesen Einstellungen finden Sie [Übernehmen einer Rolle](#) in diesem Handbuch.

Nehmen Sie die Einstellungen des Anbieters für Anmeldeinformationen an

Konfigurieren Sie diese Funktionalität wie folgt:

**credential\_source**- Einstellung für gemeinsam genutzte AWS **config** Dateien

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

Standardwert: Keiner

Zulässige Werte:

- `Umgebung` — Gibt an, dass das SDK oder Tool Quellanmeldedaten aus den Umgebungsvariablen [AWS\\_ACCESS\\_KEY\\_ID](#) und [AWS\\_SECRET\\_ACCESS\\_KEY](#) abrufen soll.
- `Ec2 InstanceMetadata` — Gibt an, dass das SDK oder Tool die dem EC2-Instanzprofil [zugeordnete IAM-Rolle verwenden soll, um Quellanmeldedaten](#) abzurufen.
- `EcsContainer` — Gibt an, dass das SDK oder Tool die dem [ECS-Container zugeordnete IAM-Rolle verwenden soll, um Quellanmeldedaten](#) abzurufen.

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

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

```
credential_source = Ec2InstanceMetadata
```

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

### **duration\_seconds**- Einstellung für gemeinsam genutzte AWS **config** Dateien

Gibt die maximale Dauer der Rollensitzung in Sekunden an.

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

Standardwert: 3600 Sekunden (eine Stunde)

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

Beispiel für die Einstellung dieser Einstellung in einer config Datei:

```
duration_seconds = 43200
```

### **external\_id**- Einstellung für gemeinsam genutzte AWS **config** Dateien

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

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

Standardwert: Keiner.

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

Beispiel für die Einstellung in einer config Datei:

```
external_id = unique_value_assigned_by_3rd_party
```

### **mfa\_serial**- Einstellung für gemeinsam genutzte AWS **config** Dateien

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

Erforderlich, wenn Sie eine Rolle übernehmen, bei der die Vertrauensrichtlinie für diese Rolle eine Bedingung beinhaltet, die eine MFA-Authentifizierung erfordert.

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. Weitere Informationen zu MFA finden Sie unter [Konfiguration des MFA-geschützten API-Zugriffs](#) im IAM-Benutzerhandbuch.

Beispiel für die Einstellung in einer Datei: `config`

```
mfa_serial = arn:aws:iam::123456789012:mfa/my-user-name
```

**role\_arn**- Einstellung für gemeinsam genutzte AWS **config** Dateien, **AWS\_IAM\_ROLE\_ARN**-Umgebungsvariable, **aws.roleArn**- JVM-Systemeigenschaft: Nur Java/Kotlin

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

Standardwert: Keiner.

Gültige Werte: Der Wert muss der ARN einer IAM-Rolle sein 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 einem Amazon EC2 EC2-Instance-Profil zugeordnet sind, oder einer Amazon ECS-Container-Instance.
- **web\_identity\_token\_file**— Um öffentliche Identitätsanbieter oder einen OpenID Connect (OIDC) -kompatiblen Identitätsanbieter für Benutzer zu verwenden, die in einer Mobil- oder Webanwendung authentifiziert wurden.

**role\_session\_name**- Einstellung für gemeinsam genutzte Dateien AWS **config**,  
**AWS\_IAM\_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.

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

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

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

Beispiel für die Einstellung in einer `config` Datei:

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

**source\_profile**- Einstellung für gemeinsam genutzte AWS **config** Dateien

Gibt ein anderes Profil an, dessen Anmeldeinformationen verwendet werden, um die Rolle anzunehmen, die in der `role_arn` Einstellung im ursprünglichen Profil angegeben ist. Informationen zur Verwendung von Profilen in geteilten `credentials` Dateien AWS `config` und Dateien finden Sie 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 das SDK auf ein Profil mit Anmeldeinformationen trifft. Die Rollenverkettung begrenzt Ihre Rollensitzung AWS CLI oder Ihre AWS API-Rollensitzung auf maximal eine Stunde und kann nicht verlängert werden. Weitere Informationen finden Sie im IAM-Benutzerhandbuch unter [Begriffe und Konzepte für Rollen](#).

Standardwert: Keiner.





Gültige Werte: 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

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

SDK	U	Hinweise oder weitere Informationen
<a href="#">AWS CLI v2</a>	Ja	
<a href="#">SDK for C++</a>	Teilwe	<code>credential_source</code> wird nicht unterstützt. <code>duration_seconds</code> nicht unterstützt. <code>mfa_serial</code> nicht unterstützt.
<a href="#">SDK for Go V2 (1.x)</a>	Ja	
<a href="#">SDK for Go 1.x (V1)</a>	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 <a href="#">Sessions</a> .
<a href="#">SDK for Java 2.x</a>	Teilwe	<code>mfa_serial</code> wird nicht unterstützt. Verwenden Sie <code>AWS_ROLE_ARN</code> anstelle von <code>AWS_IAM_ROLE_ARN</code> . Verwenden Sie <code>AWS_ROLE_SESSION_NAME</code> statt <code>AWS_IAM_ROLE_SESSION_NAME</code> .
<a href="#">SDK for Java 1.x</a>	Teilwe	<code>mfa_serial</code> wird nicht unterstützt. JVM-Systemeigenschaften werden nicht unterstützt.
<a href="#">SDK für 3.x JavaScript</a>	Ja	
<a href="#">SDK für 2.x JavaScript</a>	Teilwe	<code>credential_source</code> wird nicht unterstützt.

SDK	U zt	Hinweise oder weitere Informationen
<a href="#">SDK für Kotlin</a>	Ja	Verwenden Sie <code>AWS_ROLE_ARN</code> statt <code>AWS_IAM_ROLE_ARN</code> . Verwenden Sie <code>AWS_ROLE_SESSION_NAME</code> statt <code>AWS_IAM_ROLE_SESSION_NAME</code> .
<a href="#">SDK for .NET 3.x</a>	Ja	
<a href="#">SDK for PHP 3.x</a>	Ja	
<a href="#">SDK for Python (Boto3)</a>	Ja	
<a href="#">SDK for Ruby 3.x</a>	Ja	
<a href="#">SDK für Rust</a>	Ja	
<a href="#">Tools für PowerShell</a>	Ja	

## Anbieter von Container-Anmeldeinformationen

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

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

Wenn Sie Amazon EKS verwenden, empfehlen wir Ihnen, Amazon EKS Pod Identity zu verwenden, um die Isolierung von Anmeldeinformationen, die geringsten Rechte, die Überprüfbarkeit, den unabhängigen Betrieb, die Wiederverwendbarkeit und die Skalierbarkeit zu verbessern. Sowohl Ihre Pod- als auch eine IAM-Rolle sind mit einem Kubernetes-Servicekonto verknüpft, um die Anmeldeinformationen für Ihre Anwendungen zu verwalten.

Weitere Informationen zu Amazon EKS Pod Identity finden Sie unter [Amazon EKS Pod Identities](#) im Amazon EKS-Benutzerhandbuch. Nach der Konfiguration legt Amazon EKS die Umgebungsvariablen `AWS_CONTAINER_CREDENTIALS_FULL_URI` und die `AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE` Umgebungsvariablen fest, die die SDKs und Tools zum Abrufen von Anmeldeinformationen verwenden. Informationen zur Einrichtung finden Sie unter [Einrichten des Amazon EKS Pod Identity Agent](#) im Amazon EKS-Benutzerhandbuch oder [Amazon EKS Pod Identity vereinfacht IAM-Berechtigungen für Anwendungen auf Amazon EKS-Clustern](#) auf der AWS Blog-Website.

Konfigurieren Sie diese Funktionalität wie folgt:

### **AWS\_CONTAINER\_CREDENTIALS\_FULL\_URI**- Umgebungsvariable

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

Standardwert: Keiner.

Gültige Werte: Gültiger URI.

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

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

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

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

Standardwert: Keiner.

Gültige Werte: Gültiger relativer URI.

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

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

### **AWS\_CONTAINER\_AUTHORIZATION\_TOKEN**- Umgebungsvariable

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

Standardwert: Keiner.

Gültige Werte: Zeichenfolge.

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

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

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

### **AWS\_CONTAINER\_AUTHORIZATION\_TOKEN\_FILE**- Umgebungsvariable

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

Standardwert: Keiner.

Gültige Werte: Zeichenfolge.

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

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

## Kompatibilität mit SDKs AWS

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

SDK	U zt	Hinweise oder weitere Informationen
<a href="#">AWS CLI v2</a>	Ja	
<a href="#">SDK for C++</a>	Ja	
<a href="#">SDK for Go V2 (1.x)</a>	Ja	
<a href="#">SDK for Go 1.x (V1)</a>	Ja	
<a href="#">SDK for Java 2.x</a>	Ja	
<a href="#">SDK for Java 1.x</a>	Teilwe	Amazon EKS Pod Identity und wird AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE nicht unterstützt.
<a href="#">SDK für JavaScript 3.x</a>	Ja	
<a href="#">SDK für 2.x JavaScript</a>	Ja	
<a href="#">SDK für Kotlin</a>	Ja	
<a href="#">SDK for .NET 3.x</a>	Ja	
<a href="#">SDK for PHP 3.x</a>	Ja	
<a href="#">SDK for Python (Boto3)</a>	Ja	
<a href="#">SDK for Ruby 3.x</a>	Ja	
<a href="#">SDK für Rust</a>	Ja	
<a href="#">Tools für PowerShell</a>	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-Anmeldeinformationsanbieter bezeichnet.

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

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

- Konfiguration des SSO-Token-Anbieters (empfohlen) — Verlängerte Sitzungsdauer.
- Legacy-Konfiguration, die nicht aktualisiert werden kann — Verwendet eine feste, achtstündige Sitzung.

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

Um benutzerdefinierte Sitzungsdauern festzulegen, müssen Sie die Konfiguration des SSO-Token-Anbieters verwenden.

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

- [AWS IAM Identity Center Benutzerhandbuch](#)
- [AWS IAM Identity Center Referenz zur Portal-API](#)

## Voraussetzungen

Sie müssen zuerst IAM Identity Center aktivieren. Einzelheiten zur Aktivierung der IAM Identity Center-Authentifizierung finden Sie unter [Erste Schritte](#) im AWS IAM Identity Center Benutzerhandbuch.

Folgen Sie alternativ den [Authentifizierung von IAM Identity Center](#) Anweisungen in diesem Handbuch. Diese Anweisungen bieten eine vollständige Anleitung, von der Aktivierung von IAM

Identity Center bis hin zur Konfiguration der erforderlichen gemeinsamen config Datei, die im Folgenden beschrieben wird.

## Konfiguration des SSO-Token-Anbieters

### Note

Informationen AWS CLI zum Erstellen dieser Konfiguration für Sie finden Sie unter [Konfigurieren Ihres Profils mit dem aws configure sso Assistenten](#) im AWS CLI.

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

Der `sso-session` Abschnitt der config Datei wird verwendet, um Konfigurationsvariablen für den Erwerb von SSO-Zugriffstoken zu gruppieren, die dann zum Abrufen von AWS Anmeldeinformationen verwendet werden können. Weitere Informationen zum Formatieren von Abschnitten in einer config Datei finden Sie unter [Format der Konfigurationsdatei](#).

Sie definieren einen `sso-session`-Abschnitt und ordnen ihn einem Profil zu. `sso_region` und `sso_start_url` müssen innerhalb des `sso-session`-Abschnitts festgelegt werden. Normalerweise `sso_account_id` und `sso_role_name` muss in dem `profile` Abschnitt festgelegt werden, damit das SDK AWS Anmeldeinformationen anfordern kann.

### Note

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

Im folgenden Beispiel wird das SDK so konfiguriert, dass es IAM Identity Center-Anmeldeinformationen anfordert. Es unterstützt auch die automatische Token-Aktualisierung.

```
[profile dev]
```

```
sso_session = my-ss0
sso_account_id = 111122223333
sso_role_name = SampleRole

[sso-session my-ss0]
sso_region = us-east-1
sso_start_url = https://my-ss0-portal.awsapps.com/start
sso_registration_scopes = sso:account:access
```

Sie können sso-session Konfigurationen für mehrere Profile wiederverwenden.

```
[profile dev]
sso_session = my-ss0
sso_account_id = 111122223333
sso_role_name = SampleRole

[profile prod]
sso_session = my-ss0
sso_account_id = 111122223333
sso_role_name = SampleRole2

[sso-session my-ss0]
sso_region = us-east-1
sso_start_url = https://my-ss0-portal.awsapps.com/start
sso_registration_scopes = sso:account:access
```

sso\_account\_id und sso\_role\_name sind nicht für alle Szenarien der SSO-Token-Konfiguration erforderlich. Wenn Ihre Anwendung nur AWS-Services diese Unterstützung für die Trägerauthentifizierung verwendet, sind herkömmliche AWS Anmeldeinformationen nicht erforderlich. Bei der Bearer-Authentifizierung handelt es sich um ein HTTP-Authentifizierungsschema, das Sicherheitstoken, sogenannte Bearer-Token, verwendet. In diesem Szenario sind sso\_account\_id und sso\_role\_name nicht erforderlich. Sehen Sie in der jeweiligen Anleitung AWS-Service nach, ob sie die Bearer-Token-Autorisierung unterstützt.

Registrierungsbereiche werden als Teil eines konfiguriert. sso-session Der Geltungsbereich ist ein Mechanismus OAuth 2.0, mit dem der Zugriff einer Anwendung auf das Konto eines Benutzers beschränkt wird. Eine Anwendung kann einen oder mehrere Bereiche anfordern, und das für die Anwendung ausgegebene Zugriffstoken ist auf die zugewiesenen Bereiche beschränkt. Diese Bereiche definieren die Berechtigungen, die für die Autorisierung für den registrierten OIDC-Client angefordert werden, und die vom Client abgerufenen Zugriffstoken. Informationen zu den unterstützten Zugriffsbereichsoptionen finden Sie unter [Zugriffsbereiche](#) im AWS IAM Identity Center



Benutzerhandbuch. Im folgenden Beispiel wird der `sso_registration_scopes` Zugriff für die Auflistung von Konten und Rollen bereitgestellt.

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

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

## Nicht aktualisierbare Legacy-Konfiguration

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

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

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

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

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

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

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

## Einstellungen des IAM Identity Center-Anmeldeinformationsanbieters

Konfigurieren Sie diese Funktionalität wie folgt:

### **sso\_start\_url**- Einstellung für gemeinsam genutzte AWS **config** Dateien

Die URL, die auf das IAM Identity Center-Zugriffportal Ihrer Organisation verweist. Weitere Informationen zum IAM Identity Center-Zugriffportal finden Sie im AWS IAM Identity Center Benutzerhandbuch [unter Verwenden des AWS Zugriffsportals](#).

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

### **sso\_region**- Einstellung für gemeinsam genutzte AWS **config** Dateien

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

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

### **sso\_account\_id**- Einstellung für gemeinsam genutzte AWS **config** Dateien

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

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

### **sso\_role\_name**- Einstellung für gemeinsam genutzte Dateien AWS **config**

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

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

Um die Liste der verfügbaren Berechtigungssätze pro zu sehen AWS-Konto, gehen Sie zur [IAM Identity Center-Konsole](#) und öffnen Sie die AWS-KontenSeite. Wählen Sie den richtigen Namen für den Berechtigungssatz aus, der in der AWS-Konten Tabelle aufgeführt ist. Die Liste der verfügbaren Berechtigungssätze, die die [ListAccountRoles](#)API-Methode verwenden, finden Sie auch in der AWS IAM Identity Center Portal-API-Referenz. 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` Bereiche autorisieren den Zugriff auf über IAM-Identity-Center-Bearer-Token autorisierte Endpunkte. Ein Mindestbereich von `sso:account:access` muss gewährt werden, um ein Aktualisierungstoken vom IAM Identity Center-Dienst zurückzuerhalten. Die unterstützten [Zugriffsbereichszeichenfolgen finden Sie unter Zugriffsbereiche](#) im AWS IAM Identity Center Benutzerhandbuch. 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 SDKs AWS

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

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

SDK	U zt	Hinweise oder weitere Informationen
<a href="#">SDK for Java 1.x</a>	Nein	
<a href="#">SDK für 3.x JavaScript</a>	Ja	
<a href="#">SDK für 2.x JavaScript</a>	Ja	
<a href="#">SDK für Kotlin</a>	Ja	
<a href="#">SDK for .NET 3.x</a>	Ja	
<a href="#">SDK for PHP 3.x</a>	Ja	
<a href="#">SDK for Python (Boto3)</a>	Ja	
<a href="#">SDK for Ruby 3.x</a>	Ja	
<a href="#">SDK für Rust</a>	Teilwe	Nur ältere, nicht aktualisierbare Konfiguration.
<a href="#">Tools für PowerShell</a>	Ja	

## IMDS-Anmeldeinformationsanbieter

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 [Instance-Metadaten und Benutzerdaten](#) im Amazon EC2 EC2-Benutzerhandbuch oder [Instance-Metadaten und Benutzerdaten](#) im Amazon EC2 EC2-Benutzerhandbuch. Amazon EC2 bietet einen lokalen Endpunkt, der Instances zur Verfügung stellt und der Instance verschiedene Informationen zur Verfügung stellen kann. Wenn der Instance eine Rolle zugewiesen ist, kann sie eine Reihe von Anmeldeinformationen bereitstellen, die für diese Rolle gültig sind. Die SDKs können diesen Endpunkt verwenden, um Anmeldeinformationen als Teil ihrer [standardmäßigen Anbieterkette für Anmeldeinformationen aufzulösen](#). Standardmäßig wird Instance Metadata Service Version 2 (IMDSv2) verwendet, eine sicherere Version von IMDS, die ein Sitzungstoken verwendet. Wenn dies aufgrund eines Zustands fehlschlägt, der nicht erneut versucht werden kann (HTTP-Fehlercodes 403, 404, 405), wird IMDSv1 als Fallback verwendet.

Konfigurieren Sie diese Funktionalität wie folgt:

## **AWS\_EC2\_METADATA\_DISABLED**- Umgebungsvariable

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

Standardwert: `false`.

Zulässige Werte:

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

**ec2\_metadata\_v1\_disabled**- Einstellung für gemeinsam genutzte AWS **config** Dateien, **AWS\_EC2\_METADATA\_V1\_DISABLED**- Umgebungsvariable, **aws.disableEc2MetadataV1**- JVM-Systemeigenschaft: Nur Java/Kotlin

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

### Note

Neue SDKs unterstützen IMDSv1 nicht und unterstützen daher diese Einstellung nicht. Einzelheiten finden Sie in der Tabelle. [Kompatibilität mit SDKs AWS](#)

Standardwert: `false`.

Zulässige Werte:

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

**ec2\_metadata\_service\_endpoint**- Einstellung für gemeinsam genutzte Dateien **AWS config**, **AWS\_EC2\_METADATA\_SERVICE\_ENDPOINT**- Umgebungsvariable, **aws.ec2MetadataServiceEndpoint**- JVM-Systemeigenschaft: Nur Java/Kotlin

Der Endpunkt von IMDS.

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: Gültiger URI.

**ec2\_metadata\_service\_endpoint\_mode**- Einstellung für gemeinsam genutzte AWS **config** Dateien, **AWS\_EC2\_METADATA\_SERVICE\_ENDPOINT\_MODE**- Umgebungsvariable, **aws.ec2MetadataServiceEndpointMode**- JVM-Systemeigenschaft: Nur Java/Kotlin

Der Endpunktmodus von IMDS.

Standardwert:IPv4.

Gültige Werte:IPv4,IPv6.

### Note

Der IMDS-Anmeldeinformationsanbieter ist Teil von. [Kette von Anbietern von Anmeldeinformationen](#) Der IMDS-Anbieter für Anmeldeinformationen wird jedoch erst nach mehreren anderen Anbietern dieser Serie überprüft. Wenn Sie also möchten, dass Ihr Programm die Anmeldeinformationen dieses Anbieters verwendet, müssen Sie andere gültige Anmeldeinformationsanbieter aus Ihrer Konfiguration entfernen oder ein anderes Profil verwenden. Anstatt sich auf die Kette der Anmeldeinformationsanbieter zu verlassen, um automatisch zu ermitteln, welcher Anbieter gültige Anmeldeinformationen zurückgibt, können Sie alternativ die Verwendung des IMDS-Anmeldeinformationsanbieters im Code angeben. Sie können die Quellen für Anmeldeinformationen direkt angeben, wenn Sie Service-Clients erstellen.

## Sicherheit für IMDS-Anmeldeinformationen

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

### Note

AWS SDK-Clients, die mit gültigen Anmeldeinformationen konfiguriert sind, verwenden unabhängig von diesen Einstellungen niemals IMDS zum Abrufen von Anmeldeinformationen.

## Verwendung von Amazon EC2 IMDS-Anmeldeinformationen deaktivieren

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

### Unter Linux und macOS

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

```
$ export AWS_EC2_METADATA_DISABLED=true
```

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

### Windows

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

```
$ set AWS_EC2_METADATA_DISABLED=true
```

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

```
$ setx AWS_EC2_METADATA_DISABLED=true
```

#### Note

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

## Kompatibilität mit SDKs AWS

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

SDK	U zt	Hinweise oder weitere Informationen
<a href="#">AWS CLI v2</a>	Ja	
<a href="#">SDK for C++</a>	Ja	
<a href="#">SDK for Go V2 (1.x)</a>	Ja	
<a href="#">SDK for Go 1.x (V1)</a>	Ja	Um die Einstellungen für gemeinsam genutzte config Dateien zu verwenden, müssen Sie das Laden aus der Konfigurationsdatei aktivieren. Weitere Informationen finden Sie unter <a href="#">Sessions</a> .
<a href="#">SDK for Java 2.x</a>	Ja	
<a href="#">SDK for Java 1.x</a>	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.
<a href="#">SDK für 3.x JavaScript</a>	Ja	
<a href="#">SDK für 2.x JavaScript</a>	Ja	
<a href="#">SDK für Kotlin</a>	Ja	Verwendet kein IMDSv1-Fallback.
<a href="#">SDK for .NET 3.x</a>	Ja	
<a href="#">SDK for PHP 3.x</a>	Ja	
<a href="#">SDK for Python (Boto3)</a>	Ja	
<a href="#">SDK for Ruby 3.x</a>	Ja	
<a href="#">SDK für Rust</a>	Ja	Verwendet kein IMDSv1-Fallback.



SDK	U zt	Hinweise oder weitere Informationen
<a href="#">Tools für PowerShell</a>	Ja	Sie können den IMDSv1-Fallback explizit im Code mithilfe von deaktivieren. <code>[Amazon.Util.EC2InstanceMetadata]::EC2MetadataV1Disabled = \$true</code>

## Anbieter von Prozessanmeldeinformationen

SDKs bieten eine Möglichkeit, die Kette der Anbieter von Anmeldeinformationen für benutzerdefinierte Anwendungsfälle zu erweitern.

IAM Roles Anywhere bietet eine Möglichkeit, temporäre Anmeldeinformationen für einen Workload oder Prozess abzurufen, der außerhalb von ausgeführt wird. AWS Informationen zur Konfiguration `credential_process` für diese Verwendung finden Sie unter [IAM Roles Anywhere](#).

### Warning

Im Folgenden wird eine Methode zur Beschaffung von Anmeldeinformationen aus einem externen Prozess beschrieben. Dies kann potenziell gefährlich sein, gehen Sie also vorsichtig vor. Andere Anbieter von Anmeldeinformationen sollten nach Möglichkeit bevorzugt werden. Wenn Sie diese Option verwenden, sollten Sie sicherstellen, dass die `config` Datei so weit wie möglich gesperrt ist, indem Sie die bewährten Sicherheitsmethoden für Ihr Betriebssystem verwenden. Stellen Sie sicher, dass Ihr benutzerdefiniertes Anmeldeinformationstool keine geheimen Informationen speichert `StdErr`, da die SDKs solche Informationen erfassen und protokollieren AWS CLI können, wodurch sie möglicherweise unbefugten Benutzern zugänglich gemacht werden.

Konfigurieren Sie diese Funktionalität wie folgt:

### **credential\_process**- Einstellung für gemeinsam genutzte AWS **config** Dateien

Gibt einen externen Befehl an, den das SDK oder Tool in Ihrem Namen ausführt, um zu verwendende Authentifizierungsdaten zu generieren oder abzurufen. Die Einstellung gibt den Namen eines Programms oder Befehls an, den das SDK aufruft. Wenn das SDK den Prozess aufruft, wartet es darauf, dass der Prozess JSON-Daten in sie schreibt. `stdout` Der

benutzerdefinierte Anbieter muss Informationen in einem bestimmten Format zurückgeben. Diese Informationen enthalten die Anmeldeinformationen, mit denen das SDK oder das Tool Sie authentifizieren kann.

### Note

Der Anbieter von Prozessanmeldedaten ist Teil von. [Kette von Anbietern 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.

## Den Pfad zum Programm für Anmeldeinformationen angeben

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

- Der Pfad und der Dateiname dürfen nur aus 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.

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"
```

## Gültige Ausgabe des Credentials-Programms

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

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

### Note

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

Der Expiration Schlüssel ist ein Zeitstempel im RFC3339-Format. Wenn der Expiration Schlüssel nicht in der Ausgabe des Tools enthalten ist, geht das SDK davon aus, dass es sich bei den Anmeldeinformationen um langfristige Anmeldeinformationen handelt, die nicht aktualisiert werden. Andernfalls werden die Anmeldeinformationen als temporäre Anmeldeinformationen

betrachtet und sie werden automatisch aktualisiert, indem der `credential_process` Befehl erneut ausgeführt wird, bevor die Anmeldeinformationen ablaufen.

### Note

Das SDK speichert die Anmeldeinformationen für externe Prozesse nicht im Cache, so wie es bei der Übernahme von Rollenmeldedaten der Fall ist. 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 SDKs AWS

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

SDK	U zt	Hinweise oder weitere Informationen
<a href="#">AWS CLI v2</a>	Ja	
<a href="#">SDK for C++</a>	Ja	
<a href="#">SDK for Go V2 (1.x)</a>	Ja	
<a href="#">SDK for Go 1.x (V1)</a>	Ja	Um die Einstellungen für gemeinsam genutzte <code>config</code> Dateien zu verwenden, müssen Sie das Laden aus der Konfigurationsdatei aktivieren; siehe <a href="#">Sessions</a> .
<a href="#">SDK for Java 2.x</a>	Ja	
<a href="#">SDK for Java 1.x</a>	Ja	
<a href="#">SDK für 3.x JavaScript</a>	Ja	
<a href="#">SDK für 2.x JavaScript</a>	Ja	

SDK	U zt	Hinweise oder weitere Informationen
<a href="#">SDK für Kotlin</a>	Ja	
<a href="#">SDK for .NET 3.x</a>	Ja	
<a href="#">SDK for PHP 3.x</a>	Ja	
<a href="#">SDK for Python (Boto3)</a>	Ja	
<a href="#">SDK for Ruby 3.x</a>	Ja	
<a href="#">SDK für Rust</a>	Ja	
<a href="#">Tools für PowerShell</a>	Ja	

## AWS Standardisierte Funktionen von SDKs und Tools

Viele Funktionen wurden auf einheitliche Standardwerte standardisiert und funktionieren in vielen SDKs auf die gleiche Weise. Diese Konsistenz erhöht die Produktivität und Klarheit bei der Codierung über mehrere SDKs hinweg. 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

- [Application ID](#)
- [Amazon EC2-Instance-Metadaten](#)
- [Amazon S3 Access Points](#)
- [Multiregionale Amazon-S3-Zugriffspunkte](#)
- [AWS-Region](#)
- [AWS STS Regionalisierte Endpunkte](#)

- [Dual-Stack- und FIPS-Endpunkte](#)
- [Endpunkterkennung](#)
- [Allgemeine Konfigurationseinstellungen](#)
- [IMDS-Kunde](#)
- [Wiederholungsverhalten](#)
- [Komprimierung anfordern](#)
- [Servicespezifische Endpunkte](#)
- [Standardeinstellungen für intelligente Konfigurationen](#)

## Application ID

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

Konfigurieren Sie diese Funktionalität wie folgt:

**sdk\_ua\_app\_id**- Einstellung für gemeinsam genutzte AWS **config** Dateien,

**AWS\_SDK\_UA\_APP\_ID**- Umgebungsvariable, **aws.userAgentAppId**- JVM-Systemeigenschaft: Nur Java/Kotlin

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

Standardwert: None

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

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

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

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

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

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

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

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

## Kompatibilität mit AWS SDKs

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

SDK	U zt	Hinweise oder weitere Informationen
<a href="#">AWS CLI v2</a>	Nein	
<a href="#">SDK for C++</a>	Ja	Die gemeinsam genutzte config Datei wird nicht unterstützt.
<a href="#">SDK for Go V2 (1.x)</a>	Ja	
<a href="#">SDK for Go 1.x (V1)</a>	Nein	
<a href="#">SDK for Java 2.x</a>	Teilwe	configDie Einstellung für gemeinsam genutzte Dateien wird nicht unterstützt; die Umgebungsvariable wird nicht unterstützt.
<a href="#">SDK for Java 1.x</a>	Nein	

SDK	U zt	Hinweise oder weitere Informationen
<a href="#">SDK für 3.x JavaScript</a>	Ja	
<a href="#">SDK für 2.x JavaScript</a>	Nein	
<a href="#">SDK für Kotlin</a>	Ja	
<a href="#">SDK for .NET 3.x</a>	Ja	Umgebungsvariablen werden nicht unterstützt.
<a href="#">SDK for PHP 3.x</a>	Ja	
<a href="#">SDK for Python (Boto3)</a>	Ja	
<a href="#">SDK for Ruby 3.x</a>	Ja	
<a href="#">SDK für Rust</a>	Ja	
<a href="#">Tools für PowerShell</a>	Nein	

## Amazon EC2-Instance-Metadaten

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

Konfigurieren Sie diese Funktionalität wie folgt:

**metadata\_service\_num\_attempts**- Einstellung für gemeinsam genutzte AWS **config** Dateien, **AWS\_METADATA\_SERVICE\_NUM\_ATTEMPTS**- Umgebungsvariable

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

Standardwert: 1

Gültige Werte: Zahl größer oder gleich 1.



**metadata\_service\_timeout**- Einstellung für gemeinsam genutzte AWS **config** Dateien,  
**AWS\_METADATA\_SERVICE\_TIMEOUT**- Umgebungsvariable

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

Standardwert: 1

Gültige Werte: Zahl größer oder gleich 1.

Beispiel für das Einstellen dieser Werte in der config Datei:

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

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

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

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

```
setx AWS_METADATA_SERVICE_NUM_ATTEMPTS 10
setx AWS_METADATA_SERVICE_TIMEOUT 10
```

## Kompatibilität mit AWS SDKs

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

SDK	U zt	Hinweise oder weitere Informationen
<a href="#">AWS CLI v2</a>	Ja	
<a href="#">SDK for C++</a>	Nein	

SDK	U zt	Hinweise oder weitere Informationen
<a href="#">SDK for Go V2 (1.x)</a>	Nein	
<a href="#">SDK for Go 1.x (V1)</a>	Nein	
<a href="#">SDK for Java 2.x</a>	Nein	
<a href="#">SDK for Java 1.x</a>	Teilwe	Nur AWS_METADATA_SERVICE_TIMEOUT wird unterstützt.
<a href="#">SDK für 3.x JavaScript</a>	Nein	
<a href="#">SDK für 2.x JavaScript</a>	Nein	
<a href="#">SDK für Kotlin</a>	Nein	
<a href="#">SDK for .NET 3.x</a>	Nein	
<a href="#">SDK for PHP 3.x</a>	Ja	
<a href="#">SDK for Python (Boto3)</a>	Ja	
<a href="#">SDK for Ruby 3.x</a>	Nein	
<a href="#">SDK für Rust</a>	Nein	
<a href="#">Tools für PowerShell</a>	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 Access Point Amazon Resource Names (ARNs) im Bucket-Feld für API-Operationen verwenden, anstatt den Bucket-Namen explizit anzugeben. Sie werden für bestimmte Operationen verwendet, z. B. die Verwendung eines Access Point-ARN [GetObject](#) zum Abrufen eines Objekts aus einem Bucket oder die Verwendung eines Access Point-ARN mit, [PutObject](#) einem Bucket ein Objekt hinzuzufügen.

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

Konfigurieren Sie diese Funktionalität wie folgt:

**s3\_use\_arn\_region**- Einstellung für gemeinsam genutzte AWS **config** Dateien, **AWS\_S3\_USE\_ARN\_REGION**- Umgebungsvariable, **aws.s3UseArnRegion**- JVM-Systemeigenschaft: Nur Java/Kotlin, Um den Wert direkt im Code zu konfigurieren, wenden Sie sich direkt an Ihr spezielles SDK.

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

Standardwert: `false`

Zulässige Werte:

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

## Kompatibilität mit SDKs AWS

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

SDK	U  zt	Hinweise oder weitere Informationen
<a href="#">AWS CLI v2</a>	Ja	

SDK	Unz	Hinweise oder weitere Informationen
<a href="#">SDK for C++</a>	Ja	
<a href="#">SDK for Go V2 (1.x)</a>	Ja	
<a href="#">SDK for Go 1.x (V1)</a>	Ja	Um die Einstellungen für gemeinsam genutzte config Dateien zu verwenden, müssen Sie das Laden aus der Konfigurationsdatei aktivieren. Weitere Informationen finden Sie unter <a href="#">Sessions</a> .
<a href="#">SDK for Java 2.x</a>	Ja	
<a href="#">SDK for Java 1.x</a>	Ja	Die JVM-Systemeigenschaft wird nicht unterstützt.
<a href="#">SDK für 3.x JavaScript</a>	Ja	
<a href="#">SDK für 2.x JavaScript</a>	Ja	
<a href="#">SDK für Kotlin</a>	Ja	
<a href="#">SDK for .NET 3.x</a>	Ja	Entspricht nicht der Standardpriorität; der Wert einer gemeinsam genutzten config Datei hat Vorrang vor der Umgebungsvariablen.
<a href="#">SDK for PHP 3.x</a>	Ja	
<a href="#">SDK for Python (Boto3)</a>	Ja	
<a href="#">SDK for Ruby 3.x</a>	Ja	
<a href="#">SDK für Rust</a>	Nein	
<a href="#">Tools für PowerShell</a>	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 befinden. 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 mit einem multiregionalen 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 vom SDK durch eine Abhängigkeit von der abgerufen. [AWSCommon Runtime \(CRT\) -Bibliotheken](#)

Konfigurieren Sie diese Funktionalität wie folgt:

**s3\_disable\_multiregion\_access\_points**- Einstellung für gemeinsam genutzte AWS **config** Dateien, **AWS\_S3\_DISABLE\_MULTIREGION\_ACCESS\_POINTS**- Umgebungsvariable, **aws.s3DisableMultiRegionAccessPoints**- JVM-Systemeigenschaft: Nur Java/Kotlin, Um den Wert direkt im Code zu konfigurieren, wenden Sie sich direkt an Ihr spezielles SDK.

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

Standardwert: `false`

Zulässige Werte:

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

## Kompatibilität mit SDKs AWS

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

SDK	Unterstützt	Hinweise oder weitere Informationen
<a href="#">AWS CLI v2</a>	Ja	
<a href="#">SDK for C++</a>	Ja	
<a href="#">SDK for Go V2 (1.x)</a>	Ja	
<a href="#">SDK for Go 1.x (V1)</a>	Nein	
<a href="#">SDK for Java 2.x</a>	Ja	
<a href="#">SDK for Java 1.x</a>	Nein	
<a href="#">SDK für 3.x JavaScript</a>	Ja	
<a href="#">SDK für 2.x JavaScript</a>	Nein	
<a href="#">SDK für Kotlin</a>	Ja	
<a href="#">SDK for .NET 3.x</a>	Ja	
<a href="#">SDK for PHP 3.x</a>	Ja	
<a href="#">SDK for Python (Boto3)</a>	Ja	
<a href="#">SDK for Ruby 3.x</a>	Ja	
<a href="#">SDK für Rust</a>	Ja	
<a href="#">Tools für PowerShell</a>	Ja	

## AWS-Region

AWS-Regionen sind ein wichtiges Konzept, das man verstehen sollte, 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 laufen, wo Sie und Ihre Benutzer darauf zugreifen. Regionen bieten Fehlertoleranz, Stabilität und Ausfallsicherheit und können auch die Latenz verkürzen. Mit Regionen können Sie redundante Ressourcen einrichten, die verfügbar bleiben und von einem regionalen Ausfall nicht betroffen sind.

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

Das Allgemeine AWS-Referenzenthält Informationen zu folgenden Themen:

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

### Service-Clients erstellen

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

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

Konfigurieren Sie diese Funktionalität wie folgt:

**region**- Einstellung für gemeinsam genutzte AWS **config** Dateien, **AWS\_REGION**-Umgebungsvariable, **aws.region**- JVM-Systemeigenschaft: Nur Java/Kotlin

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

Standardwert: Keiner. Sie müssen diesen Wert explizit angeben.

Zulässige Werte:

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

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

```
[default]
region = us-west-2
```

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

```
export AWS_REGION=us-west-2
```

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

```
setx AWS_REGION us-west-2
```

Die meisten SDKs verfügen über ein „Konfiguration“-Objekt, mit dem die Standardregion im Anwendungscode festgelegt werden kann. Einzelheiten finden Sie in Ihrem spezifischen AWS SDK-Entwicklerhandbuch.



## Kompatibilität mit AWS SDKs

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

SDK	U zt	Hinweise oder weitere Informationen
<a href="#">AWS CLI v2</a>	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 überprüft).
<a href="#">AWS CLI v1</a>	Ja	AWS CLI v1 verwendet eine zu diesem <code>AWS_DEFAULT_REGION</code> Zweck benannte Umgebungsvariable.
<a href="#">SDK for C++</a>	Ja	
<a href="#">SDK for Go V2 (1.x)</a>	Ja	
<a href="#">SDK for Go 1.x (V1)</a>	Ja	Um die Einstellungen für gemeinsam genutzte <code>config</code> Dateien zu verwenden, müssen Sie das Laden aus der Konfigurationsdatei aktivieren; siehe <a href="#">Sessions</a> .
<a href="#">SDK for Java 2.x</a>	Ja	
<a href="#">SDK for Java 1.x</a>	Ja	
<a href="#">SDK für 3.x JavaScript</a>	Ja	
<a href="#">SDK für 2.x JavaScript</a>	Ja	
<a href="#">SDK für Kotlin</a>	Ja	
<a href="#">SDK for .NET 3.x</a>	Ja	
<a href="#">SDK for PHP 3.x</a>	Ja	
<a href="#">SDK for Python (Boto3)</a>	Ja	Dieses SDK verwendet eine zu diesem <code>AWS_DEFAULT_REGION</code> Zweck benannte Umgebungsvariable.

SDK	U zt	Hinweise oder weitere Informationen
<a href="#">SDK for Ruby 3.x</a>	Ja	
<a href="#">SDK für Rust</a>	Ja	
<a href="#">Tools für PowerShell</a>	Ja	

## AWS STS Regionalisierte Endpunkte

Standardmäßig ist AWS Security Token Service (AWS STS) als globaler Dienst verfügbar, und alle AWS STS Anfragen gehen an einen einzigen Endpunkt unter `https://sts.amazonaws.com`. Globale Anfragen beziehen sich auf die Region USA Ost (Nord-Virginia). AWS empfiehlt die Verwendung regionaler AWS STS Endpunkte anstelle des globalen Endpunkts. Weitere Informationen zu Endpunkten finden Sie unter AWS STS [Endpunkte](#) in der AWS Security Token Service API-Referenz.

Konfigurieren Sie diese Funktionalität wie folgt:

**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`, für die folgenden AWS Regionen: `ap-northeast-1`, `ap-south-1`, `ap-southeast-1`, `ap-southeast-2`, `aws-global`, `ca-central-1`, `eu-central-1`, `eu-north-1`, `eu-west-1`, `eu-west-2`, `eu-west-3`, `sa-east-1`, `us-east-1`, `us-east-2`, `us-west-1`, und `us-west-2`. Für alle anderen Regionen wird automatisch deren jeweiliger regionaler Endpunkt verwendet.
- **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

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

SDK	U  Hinweise oder weitere Informationen zt
<a href="#">AWS CLI v2</a>	Teilwe Der Standardwert ist <code>regional</code> .

SDK	U zt	Hinweise oder weitere Informationen
<a href="#">SDK for C++</a>	Teilwe	Umgebungsvariable und config Dateieinstellung werden nicht unterstützt. Das SDK funktioniert mit <code>regional</code> der Einstellung.
<a href="#">SDK for Go V2 (1.x)</a>	Ja	
<a href="#">SDK for Go 1.x (V1)</a>	Ja	Um die Einstellungen für gemeinsam genutzte config Dateien zu verwenden, müssen Sie das Laden aus der Konfigurationsdatei aktivieren; siehe <a href="#">Sessions</a> .
<a href="#">SDK for Java 2.x</a>	Ja	
<a href="#">SDK for Java 1.x</a>	Ja	
<a href="#">SDK für 3.x JavaScript</a>	Ja	
<a href="#">SDK für 2.x JavaScript</a>	Ja	
<a href="#">SDK für Kotlin</a>	Neir	
<a href="#">SDK for .NET 3.x</a>	Ja	
<a href="#">SDK for PHP 3.x</a>	Ja	
<a href="#">SDK for Python (Boto3)</a>	Ja	
<a href="#">SDK for Ruby 3.x</a>	Ja	
<a href="#">SDK für Rust</a>	Ja	
<a href="#">Tools für PowerShell</a>	Ja	

## Dual-Stack- und FIPS-Endpunkte

Konfigurieren Sie diese Funktionalität wie folgt:

**use\_dualstack\_endpoint**- Einstellung für gemeinsam genutzte AWS **config** Dateien, **AWS\_USE\_DUALSTACK\_ENDPOINT**- Umgebungsvariable, **aws.useDualstackEndpoint**- JVM-Systemeigenschaft: Nur Java/Kotlin

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

Standardwert: `false`

Zulässige Werte:

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

**use\_fips\_endpoint**- Einstellung für gemeinsam genutzte Dateien AWS **config**, **AWS\_USE\_FIPS\_ENDPOINT**- Umgebungsvariable, **aws.useFipsEndpoint**- JVM-Systemeigenschaft: Nur Java/Kotlin

Schaltet ein oder aus, ob das SDK oder das Tool Anfragen an FIPS-konforme Endpunkte sendet. Bei den Federal Information Processing Standards (FIPS) handelt es sich um eine Reihe von Sicherheitsanforderungen der US-Regierung für Daten und deren Verschlüsselung. Regierungsbehörden, Partner und Personen, die mit der Bundesregierung Geschäfte machen möchten, müssen sich an die FIPS-Richtlinien halten. Im Gegensatz zu AWS Standardendpunkten verwenden FIPS-Endpunkte eine TLS-Softwarebibliothek, die FIPS 140-2 entspricht. Wenn diese Einstellung aktiviert ist und kein FIPS-Endpunkt für den Dienst in Ihrem System vorhanden ist, schlägt der Anruf möglicherweise fehl. AWS-Region [AWS Servicespezifische Endpunkte](#) und die `--endpoint-url` Option zum AWS Command Line Interface Überschreiben dieser Einstellung.

Weitere Informationen zu anderen Methoden zur Angabe von AWS-Region FIPS-Endpunkten finden Sie unter [FIPS-Endpunkte](#) nach Dienst. Weitere Informationen zu Amazon Elastic Compute Cloud-Service-Endpunkten finden Sie unter [Dual-Stack-Endpunkte \(IPv4 und IPv6\) in der Amazon EC2 API-Referenz](#).

Standardwert: `false`

## Zulässige Werte:

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

## Kompatibilität mit AWS SDKs

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

SDK	Unterstützt	Hinweise oder weitere Informationen
<a href="#">AWS CLI v2</a>	Ja	
<a href="#">SDK for C++</a>	Ja	
<a href="#">SDK for Go V2 (1.x)</a>	Ja	
<a href="#">SDK for Go 1.x (V1)</a>	Ja	Um die Einstellungen für gemeinsam genutzte config Dateien zu verwenden, müssen Sie das Laden aus der Konfigurationsdatei aktivieren. Weitere Informationen finden Sie unter <a href="#">Sessions</a> .
<a href="#">SDK for Java 2.x</a>	Ja	
<a href="#">SDK for Java 1.x</a>	Nein	
<a href="#">SDK für 3.x JavaScript</a>	Ja	
<a href="#">SDK für 2.x JavaScript</a>	Ja	
<a href="#">SDK für Kotlin</a>	Ja	
<a href="#">SDK for .NET 3.x</a>	Ja	
<a href="#">SDK for PHP 3.x</a>	Ja	
<a href="#">SDK for Python (Boto3)</a>	Ja	

SDK	U zt	Hinweise oder weitere Informationen
<a href="#">SDK for Ruby 3.x</a>	Ja	
<a href="#">SDK für Rust</a>	Ja	
<a href="#">Tools für PowerShell</a>	Ja	

## Endpunkterkennung

SDKs verwenden Endpoint Discovery für den Zugriff auf Dienstendpunkte (URLs für den Zugriff auf verschiedene Ressourcen) und behalten gleichzeitig die Flexibilität, URLs AWS nach Bedarf zu ändern. Auf diese Weise kann Ihr Code automatisch neue Endpunkte erkennen. Für einige Dienste gibt es keine festen Endpunkte. Stattdessen erhalten Sie die verfügbaren Endpunkte während der Laufzeit, indem Sie eine Anfrage stellen, um zuerst die Endpunkte abzurufen. Nach dem Abrufen der verfügbaren Endpunkte verwendet der Code dann den Endpunkt, um auf andere Operationen zuzugreifen. Für Amazon Timestream stellt das SDK beispielsweise eine `DescribeEndpoints` Anfrage zum Abrufen der verfügbaren Endpunkte und verwendet diese Endpunkte dann, um bestimmte Operationen wie `createDatabase` oder `createTable` abzuschließen.

Konfigurieren Sie diese Funktionalität wie folgt:

**endpoint\_discovery\_enabled**- Einstellung für gemeinsam genutzte AWS **config** Dateien, **AWS\_ENABLE\_ENDPOINT\_DISCOVERY**- Umgebungsvariable, **aws.endpointDiscoveryEnabled**- JVM-Systemeigenschaft: Nur Java/Kotlin, Um den Wert direkt im Code zu konfigurieren, wenden Sie sich direkt an Ihr spezielles SDK.

Aktiviert oder deaktiviert die Endpunkterkennung für DynamoDB.

Endpoint Discovery ist in Timestream erforderlich und in Amazon DynamoDB optional. Diese Einstellung ist standardmäßig entweder `true` oder, `false` je nachdem, ob der Service eine Endpunkterkennung erfordert, voreingestellt. Timestream-Anfragen sind standardmäßig auf `true` und Amazon DynamoDB DynamoDB-Anfragen standardmäßig auf `false`.

Zulässige Werte:

- **true**— Das SDK sollte automatisch versuchen, einen Endpunkt für Dienste zu ermitteln, bei denen die Endpunkterkennung optional ist.

- **false**— Das SDK sollte nicht automatisch versuchen, einen Endpunkt für Dienste zu finden, bei denen die Endpunkterkennung optional ist.

## Kompatibilität mit AWS SDKs

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

SDK	U	Hinweise oder weitere Informationen
<a href="#">AWS CLI v2</a>	Ja	
<a href="#">SDK for C++</a>	Ja	
<a href="#">SDK for Go V2 (1.x)</a>	Ja	
<a href="#">SDK for Go 1.x (V1)</a>	Ja	Um die Einstellungen für gemeinsam genutzte config Dateien zu verwenden, müssen Sie das Laden aus der Konfigurationsdatei aktivieren; siehe <a href="#">Sessions</a> .
<a href="#">SDK for Java 2.x</a>	Ja	Das SDK for Java 2.x verwendet <code>AWS_ENDPOINT_DISCOVERY_ENABLED</code> für die Umgebungsvariable den Namen.
<a href="#">SDK for Java 1.x</a>	Teilwe	Die JVM-Systemeigenschaft wird nicht unterstützt.
<a href="#">SDK für 3.x JavaScript</a>	Ja	
<a href="#">SDK für 2.x JavaScript</a>	Ja	
<a href="#">SDK für Kotlin</a>	Ja	
<a href="#">SDK for .NET 3.x</a>	Ja	
<a href="#">SDK for PHP 3.x</a>	Ja	
<a href="#">SDK for Python (Boto3)</a>	Ja	
<a href="#">SDK for Ruby 3.x</a>	Ja	



SDK	U zt	Hinweise oder weitere Informationen
<a href="#">SDK für Rust</a>	Teilwe	Wird nur für Timestream unterstützt.
<a href="#">Tools für PowerShell</a>	Ja	

## Allgemeine Konfigurationseinstellungen

SDKs unterstützen einige allgemeine Einstellungen, mit denen das allgemeine SDK-Verhalten konfiguriert wird.

Konfigurieren Sie diese Funktionalität wie folgt:

**api\_versions**- Einstellung für gemeinsam genutzte AWS **config** Dateien

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

Standardwert: Keiner. (Die neueste API-Version wird vom SDK verwendet.)

Gültige Werte: Dies ist eine verschachtelte Einstellung, auf die eine oder mehrere eingerückte Zeilen folgen, die jeweils einen AWS Dienst und die zu verwendende API-Version angeben. In der Dokumentation zum AWS Dienst finden Sie Informationen darüber, welche API-Versionen verfügbar sind.

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

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

**ca\_bundle**- Einstellung für gemeinsam genutzte AWS **config** Dateien, **AWS\_CA\_BUNDLE**-Umgebungsvariable

Gibt den Pfad zu einem benutzerdefinierten Zertifikatspaket (einer Datei mit einer .pem Erweiterung) an, das beim Aufbau 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
```

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

```
export AWS_CA_BUNDLE=/dev/apps/ca-certs/cabundle-2019mar05.pem
```

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

```
setx AWS_CA_BUNDLE C:\dev\apps\ca-certs\cabundle-2019mar05.pem
```

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

Gibt an, wie Ergebnisse in den AWS CLI und anderen AWS SDKs und Tools formatiert werden.

Standardwert: json

Zulässige Werte:

- **json** – Die Ausgabe erfolgt im [JSON](#)-Format.
- **yaml** –Die Ausgabe erfolgt im [YAML](#)-Format.
- **yaml-stream** – Die Ausgabe erfolgt im [YAML](#)-Format und wird so auch gestreamt. Streaming ermöglicht eine schnellere Handhabung großer Datentypen.
- **text** – Die Ausgabe wird als mehrere Zeilen mit tabulatorgetrennten Zeichenfolgenwerten formatiert. Dies kann nützlich sein, um die Ausgabe an einen Textprozessor wie `grep`, `sed` oder `awk` zu übergeben.

- **table** – Die Ausgabe erfolgt in Form einer Tabelle mit den Zeichen +|-, um die Zellenrahmen zu bilden. Normalerweise wird die Information in einem benutzerfreundlichen Format wiedergegeben, das viel einfacher zu lesen ist als die anderen, jedoch programmatisch nicht so nützlich ist.

### **parameter\_validation**- Einstellung für gemeinsam genutzte Dateien AWS **config**

Gibt an, ob das SDK oder das Tool versucht, Befehlszeilenparameter zu überprüfen, bevor sie an den AWS Dienstendpunkt gesendet werden.

Standardwert: `true`

Zulässige Werte:

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

## Kompatibilität mit SDKs AWS

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

SDK	U zt	Hinweise oder weitere Informationen
<a href="#">AWS CLI v2</a>	Teilwe	<code>api_versions</code> wird nicht unterstützt.
<a href="#">SDK for C++</a>	Ja	
<a href="#">SDK for Go V2 (1.x)</a>	Teilwe	<code>api_versions</code> und wird <code>parameter_validation</code> nicht unterstützt.
<a href="#">SDK for Go 1.x (V1)</a>	Teilwe	<code>api_versions</code> und wird <code>parameter_validation</code> nicht unterstützt. Um die Einstellungen für gemeinsam genutzte

SDK	U zt	Hinweise oder weitere Informationen
		config Dateien zu verwenden, müssen Sie das Laden aus der Konfigurationsdatei aktivieren. Weitere Informationen finden Sie unter <a href="#">Sitzungen</a> .
<a href="#">SDK for Java 2.x</a>	Nein	
<a href="#">SDK for Java 1.x</a>	Nein	
<a href="#">SDK für 3.x JavaScript</a>	Ja	
<a href="#">SDK für 2.x JavaScript</a>	Ja	
<a href="#">SDK für Kotlin</a>	Nein	
<a href="#">SDK for .NET 3.x</a>	Nein	
<a href="#">SDK for PHP 3.x</a>	Ja	
<a href="#">SDK for Python (Boto3)</a>	Ja	
<a href="#">SDK for Ruby 3.x</a>	Ja	
<a href="#">SDK für Rust</a>	Nein	
<a href="#">Tools für PowerShell</a>	Nein	

## IMDS-Kunde

SDKs implementieren einen Client für Instance Metadata Service Version 2 (IMDSv2) mithilfe von sitzungsorientierten Anfragen. Weitere Informationen zu IMDSv2 finden Sie unter [Verwenden von IMDSv2](#) im Amazon EC2 EC2-Benutzerhandbuch oder [Verwenden von IMDSv2](#) im Amazon EC2 EC2-Benutzerhandbuch. Der IMDS-Client ist über ein Client-Konfigurationsobjekt konfigurierbar, das in der SDK-Codebasis verfügbar ist.

Konfigurieren Sie diese Funktionalität wie folgt:

**retries**- Mitglied des Client-Konfigurationsobjekts

Die Anzahl der zusätzlichen Wiederholungsversuche für jede fehlgeschlagene Anfrage.

Standardwert: 3

Gültige Werte: Zahl größer als 0.

**port**- Mitglied des Client-Konfigurationsobjekts

Der Port für den Endpunkt.

Standardwert: 80

Gültige Werte: Zahl.

**token\_ttl**- Mitglied des Client-Konfigurationsobjekts

Die TTL des Tokens.

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

Gültige Werte: Zahl.

**endpoint**- Mitglied des Client-Konfigurationsobjekts

Der Endpunkt von IMDS.

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

Gültige Werte: Gültiger URI.

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

**endpoint\_mode**- Mitglied des Client-Konfigurationsobjekts

Der Endpunktmodus von IMDS.

Standardwert: `IPv4`

Zulässige Werte: IPv4, IPv6

**http\_open\_timeout**- Mitglied des Client-Konfigurationsobjekts (Name kann variieren)

Die Anzahl der Sekunden, die darauf gewartet werden soll, dass die Verbindung geöffnet wird.

Standardwert: 1 Sekunde.

Gültige Werte: Zahl größer als 0.

**http\_read\_timeout**- Mitglied des Client-Konfigurationsobjekts (Name kann variieren)

Die Anzahl der Sekunden, für die ein Datenblock gelesen werden muss.

Standardwert: 1 Sekunde.

Gültige Werte: Zahl größer als 0.

**http\_debug\_output**- Mitglied des Client-Konfigurationsobjekts (Name kann variieren)

Legt einen Ausgabestream für das Debuggen fest.

Standardwert: Keiner.

Gültige Werte: Ein gültiger I/O-Stream, wie STDOUT.

**backoff**- Mitglied des Client-Konfigurationsobjekts (Name kann variieren)

Die Anzahl der Sekunden, die zwischen Wiederholungsversuchen oder einem vom Kunden bereitgestellten Backoff-Funktion zum Aufrufen in den Ruhezustand vergehen. Dadurch wird die standardmäßige exponentielle Backoff-Strategie außer Kraft gesetzt.

Standardwert: Variiert je nach SDK.

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

## Kompatibilität mit AWS SDKs

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

SDK	U zt	Hinweise oder weitere Informationen
<a href="#">AWS CLI v2</a>	Ja	
<a href="#">SDK for C++</a>	Nein	IMDSv2 wird nur intern verwendet. Siehe <a href="#">IMDS-Anmeldeinformationsanbieter</a> .
<a href="#">SDK for Go V2 (1.x)</a>	Ja	
<a href="#">SDK for Go 1.x (V1)</a>	Ja	
<a href="#">SDK for Java 2.x</a>	Ja	
<a href="#">SDK for Java 1.x</a>	Ja	
<a href="#">SDK für 3.x JavaScript</a>	Ja	
<a href="#">SDK für 2.x JavaScript</a>	Ja	
<a href="#">SDK für Kotlin</a>	Ja	
<a href="#">SDK for .NET 3.x</a>	Ja	
<a href="#">SDK for PHP 3.x</a>	Ja	
<a href="#">SDK for Python (Boto3)</a>	Ja	
<a href="#">SDK for Ruby 3.x</a>	Ja	
<a href="#">SDK für Rust</a>	Ja	
<a href="#">Tools für PowerShell</a>	Ja	

## Wiederholungsverhalten

Das Wiederholungsverhalten umfasst Einstellungen, die festlegen, wie die SDKs versuchen, nach Fehlern aufgrund von Anfragen an eine Wiederherstellung zu versuchen. AWS-Services

Konfigurieren Sie diese Funktionalität wie folgt:

**max\_attempts**- Einstellung für gemeinsam genutzte AWS **config** Dateien, **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 Ihr SDK spezifischen Standardwert (den `max_attempts` Standardwert finden Sie in Ihrem spezifischen SDK-Handbuch oder in der Codebasis Ihres SDK).
- Falls `retry_mode` ja `standard` — Unternimmt drei Versuche.
- Falls `retry_mode` ja `adaptive` — Führt drei Versuche durch.

Gültige Werte: Zahl größer als 0.

**retry\_mode**- Einstellung für gemeinsam genutzte AWS **config** Dateien, **AWS\_RETRY\_MODE**-Umgebungsvariable, **aws.retryMode**- JVM-Systemeigenschaft: Nur Java/Kotlin

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

Standardwert: `legacy` ist die Standardstrategie für Wiederholungsversuche.

Zulässige Werte:

- `legacy`— Spezifisch für Ihr SDK (lesen Sie in Ihrem spezifischen SDK-Handbuch oder in der Codebasis Ihres SDK nach).
- `standard`— Der Standardsatz von Wiederholungsregeln für alle AWS SDKs. Dieser Modus umfasst einen Standardsatz von Fehlern, die wiederholt werden, und Unterstützung für Wiederholungsquoten. Die standardmäßige maximale Anzahl von Versuchen in diesem Modus beträgt drei, sofern nicht ausdrücklich `max_attempts` konfiguriert.
- `adaptive`— Ein experimenteller Wiederholungsmodus, der die Funktionalität des Standardmodus beinhaltet, aber auch automatische clientseitige Drosselung beinhaltet. Da es sich bei diesem Modus um einen experimentellen Modus handelt, könnte er das Verhalten in future ändern.

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 Ressource verzögern, wenn Sie denselben Client für beide Ressourcen verwenden.

Standard	Adaptiv
Anwendungsfälle: Alle.	Anwendungsfälle für Anwendungen: <ol style="list-style-type: none"> <li>1. Unempfindlich gegenüber Latenz.</li> <li>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.</li> </ol>
Unterstützt Circuit-Breaking, um zu verhindern, dass das SDK es bei Ausfällen erneut versucht.	Unterstützt Circuit-Breaking, um zu verhindern, dass das SDK es bei Ausfällen erneut versucht.
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. Eine Ressource ist in diesem Fall 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, lesen Sie in der spezifischen AWS Dokumentation des von Ihnen verwendeten Dienstes nach.

## Einzelheiten zur Implementierung des Wiederholungsmodus

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

```
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:**

Token-Buckets werden nur im `adaptive` Wiederholungsmodus verwendet. Token-Buckets setzen eine maximale Anforderungsrate durch, indem sie verlangen, dass ein Token verfügbar ist, um eine Anfrage zu initiieren. Der SDK-Client kann so konfiguriert werden, dass die Anfrage entweder schnell fehlschlägt oder blockiert wird, bis ein Token verfügbar ist.

Bei der clientseitigen Ratenbegrenzung handelt es sich um einen Algorithmus, mit dem Anfragen zunächst in beliebiger Geschwindigkeit gestellt werden können, bis die Token-Zulage erreicht ist.

Sobald jedoch eine gedrosselte Antwort erkannt wird, wird der Client `rate-of-request` entsprechend eingeschränkt. Die Token-Zulage wird ebenfalls entsprechend erhöht, wenn erfolgreiche Antworten eingehen.

Mit adaptiver Ratenbegrenzung können SDKs die Geschwindigkeit, mit der Anfragen gesendet werden, verlangsamen, um der Kapazität von AWS-Services besser gerecht zu werden.

### **SendHTTPRequest:**

Die meisten AWS SDKs verwenden eine HTTP-Bibliothek, die Verbindungspools verwendet, sodass Sie eine bestehende Verbindung wiederverwenden können, wenn Sie eine HTTP-Anfrage stellen. Im Allgemeinen werden Verbindungen aufgrund von Drosselungsfehlern wiederverwendet, wenn Anfragen erneut versucht werden. Anfragen werden bei Wiederholungsversuchen aufgrund vorübergehender Fehler nicht wiederverwendet.

### **RequestBookkeeping:**

Das Wiederholungskontingent sollte aktualisiert werden, wenn die Anfrage erfolgreich ist. Nur im `adaptive` Wiederholungsmodus `maxsendrate` wird die Statusvariable basierend auf der Art der empfangenen Antwort aktualisiert.

### **Retryable:**

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

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

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

### **MAX\_ATTEMPTS:**

Wird durch die `config` Dateieinstellung oder die Umgebungsvariable angegeben.

## HasRetryQuota

In diesem Schritt werden Wiederholungsanforderungen eingeschränkt, da ein Token im Quotenbereich für Wiederholungsversuche verfügbar sein muss. Kontingentgruppen für Wiederholungsversuche sind ein Mechanismus, um Wiederholungsversuche zu verhindern, deren Erfolg unwahrscheinlich ist. Diese Kontingente hängen vom SDK, oft vom Client und manchmal sogar von den Dienstendpunkten ab. Die verfügbaren Quota-Token für Wiederholungsversuche werden entfernt, wenn Anfragen aus verschiedenen Gründen fehlschlagen, und wieder aufgefüllt, wenn sie erfolgreich sind. Wenn keine Token mehr vorhanden sind, wird die Wiederholungsschleife beendet.

## ExponentialBackoff

Bei einem Fehler, der wiederholt werden kann, wird die Wiederholungsverzögerung anhand eines verkürzten exponentiellen Backoffs berechnet. Die SDKs verwenden abgeschnittenes binäres exponentielles Backoff mit Jitter. Der folgende Algorithmus zeigt, wie die Zeit bis zum Ruhezustand (in Sekunden) für eine Antwort auf Anfrage definiert wird:  $i$

$$\text{seconds\_to\_sleep\_}i = \min(b \cdot r^i, \text{MAX\_BACKOFF})$$

Im vorherigen Algorithmus gelten die folgenden Werte:

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

$r = 2$

$\text{MAX\_BACKOFF} = 20$  seconds für die meisten SDKs. Weitere Informationen finden Sie in Ihrem spezifischen SDK-Leitfaden oder Quellcode.

## Kompatibilität mit AWS SDKs

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

SDK	U zt	Hinweise oder weitere Informationen
<a href="#">AWS CLI v2</a>	Ja	
<a href="#">SDK for C++</a>	Ja	

SDK	U zt	Hinweise oder weitere Informationen
<a href="#">SDK for Go V2 (1.x)</a>	Ja	
<a href="#">SDK for Go 1.x (V1)</a>	Nein	
<a href="#">SDK for Java 2.x</a>	Ja	
<a href="#">SDK for Java 1.x</a>	Ja	JVM-Systemeigenschaften: anstelle von verwenden <code>aws.maxAttempts</code> ; <code>com.amazonaws.sdk.maxAttempts</code> anstelle von verwenden <code>com.amazonaws.sdk.retryMode</code> . <code>aws.retryMode</code>
<a href="#">SDK für 3.x JavaScript</a>	Ja	
<a href="#">SDK für 2.x JavaScript</a>	Nein	Unterstützt eine maximale Anzahl von Wiederholungsversuchen, exponentielles Backoff mit Jitter und eine Option für eine benutzerdefinierte Methode für Wiederholungs-Backoff.
<a href="#">SDK für Kotlin</a>	Ja	
<a href="#">SDK for .NET 3.x</a>	Ja	
<a href="#">SDK for PHP 3.x</a>	Ja	
<a href="#">SDK for Python (Boto3)</a>	Ja	
<a href="#">SDK for Ruby 3.x</a>	Ja	
<a href="#">SDK für Rust</a>	Ja	
<a href="#">Tools für PowerShell</a>	Ja	

## Komprimierung anfordern

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 kann die Anzahl erfolgloser Anfragen aufgrund von Einschränkungen der Payloadgröße durch den Dienst 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**— Deaktiviert die Anforderungskomprimierung.
- **false**— Verwenden Sie nach Möglichkeit die Anforderungskomprimierung.

**request\_min\_compression\_size\_bytes**- Einstellung für gemeinsam genutzte AWS **config** Dateien, **AWS\_REQUEST\_MIN\_COMPRESSION\_SIZE\_BYTES**- Umgebungsvariable, **aws.requestMinCompressionSizeBytes**- JVM-Systemeigenschaft: Nur Java/Kotlin

Legt die Mindestgröße 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 SDKs AWS

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

SDK	Unterstützt	Hinweise oder weitere Informationen
<a href="#">AWS CLI v2</a>	Ja	
<a href="#">SDK for C++</a>	Ja	
<a href="#">SDK for Go V2 (1.x)</a>	Ja	
<a href="#">SDK for Go 1.x (V1)</a>	Nein	
<a href="#">SDK for Java 2.x</a>	Ja	
<a href="#">SDK for Java 1.x</a>	Nein	
<a href="#">SDK für 3.x JavaScript</a>	Ja	
<a href="#">SDK für 2.x JavaScript</a>	Nein	
<a href="#">SDK für Kotlin</a>	Ja	
<a href="#">SDK for .NET 3.x</a>	Ja	
<a href="#">SDK for PHP 3.x</a>	Ja	
<a href="#">SDK for Python (Boto3)</a>	Ja	
<a href="#">SDK for Ruby 3.x</a>	Ja	
<a href="#">SDK für Rust</a>	Ja	
<a href="#">Tools für PowerShell</a>	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 in festgelegten globalen Endpunkten.

`AWS_ENDPOINT_URL`

Standardwert: none



Gültige Werte: Eine URL, die das Schema und den Host für den Endpunkt enthält. Die URL kann optional eine Pfadkomponente enthalten, die ein oder mehrere Pfadsegmente enthält.

**ignore\_configured\_endpoint\_urls**- Einstellung für gemeinsam genutzte AWS **config** Dateien, **AWS\_IGNORE\_CONFIGURED\_ENDPOINT\_URLS**- Umgebungsvariable, **aws.ignoreConfiguredEndpointUrls**- JVM-Systemeigenschaft: Nur Java/Kotlin

Diese Einstellung wird verwendet, um alle benutzerdefinierten Endpunktkonfigurationen zu ignorieren.

Beachten Sie, dass jeder explizite Endpunkt, der im Code oder auf einem Service-Client selbst festgelegt ist, unabhängig von dieser Einstellung verwendet wird. Wenn Sie beispielsweise den `--endpoint-url` Befehlszeilenparameter in einen AWS CLI Befehl aufnehmen oder eine Endpunkt-URL an einen Client-Konstruktor übergeben, ist dies immer wirksam.

Standardwert: `false`

Zulässige Werte:

- **true**— Das SDK oder Tool liest keine benutzerdefinierten Konfigurationsoptionen aus der gemeinsam genutzten `config` Datei oder aus Umgebungsvariablen zum Setzen einer Endpunkt-URL.
- **false**— Das SDK oder Tool verwendet alle verfügbaren, vom Benutzer bereitgestellten Endpunkte aus der gemeinsam genutzten `config` Datei oder aus Umgebungsvariablen.

## Konfigurieren Sie Endpunkte mithilfe von Umgebungsvariablen

Um Anfragen für alle Dienste an eine benutzerdefinierte Endpunkt-URL weiterzuleiten, legen Sie die `AWS_ENDPOINT_URL` globale Umgebungsvariable fest.

```
export AWS_ENDPOINT_URL=http://localhost:4567
```

Verwenden Sie die `AWS_ENDPOINT_URL_<SERVICE>` Umgebungsvariable AWS-Service , um Anfragen für eine bestimmte URL an einen benutzerdefinierten Endpunkt weiterzuleiten. Amazon DynamoDB hat ein `serviceId` von [DynamoDB](#). Für diesen Service lautet die Umgebungsvariable für die Endpunkt-URL `AWS_ENDPOINT_URL_DYNAMODB`. Dieser Endpunkt hat Vorrang vor dem globalen Endpunkt, der `AWS_ENDPOINT_URL` für diesen Dienst eingerichtet wurde.

```
export AWS_ENDPOINT_URL_DYNAMODB=http://localhost:5678
```

Als weiteres Beispiel AWS Elastic Beanstalk hat er ein `serviceId` von [Elastic Beanstalk](#). Der AWS-Service Bezeichner basiert auf dem API-Modell, indem alle Leerzeichen `serviceId` durch Unterstriche ersetzt und alle Buchstaben in Großbuchstaben geschrieben werden. Um den Endpunkt für diesen Dienst festzulegen, lautet die entsprechende Umgebungsvariable `AWS_ENDPOINT_URL_ELASTIC_BEANSTALK`. Eine Liste aller servicespezifischen Umgebungsvariablen finden Sie unter [Identifikatoren für dienstspezifische Endpunkte](#).

```
export AWS_ENDPOINT_URL_ELASTIC_BEANSTALK=http://localhost:5567
```

## Konfigurieren Sie Endpunkte mithilfe 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 die dienstspezifischen Endpunkt-URLs für Amazon S3 und AWS Elastic Beanstalk im selben Profil festgelegt werden. AWS Elastic Beanstalk hat einen `serviceId` von [Elastic Beanstalk](#). Der AWS-Service Bezeichner basiert auf dem API-Modell, `serviceId` indem alle Leerzeichen durch Unterstriche ersetzt und alle Buchstaben klein geschrieben werden. Somit wird der Service-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 B verwenden und in Ihrem Code Amazon EC2 aufrufen, wird der Endpunkt als `https://profile-b-ec2-endpoint.aws` aufgelöst. Wenn Ihr Code eine Anforderung für einen anderen Service stellt, folgt die Endpunktauflösung keiner benutzerdefinierten Logik. Der Endpunkt wird nicht zu dem im Profil A definierten globalen Endpunkt aufgelöst. Damit ein globaler Endpunkt für das Profil B wirksam wird, müssten Sie `endpoint_url` direkt im Profil B festlegen. Weitere Informationen zur `source_profile`-Einstellung finden Sie unter [Übernehmen Sie die Rolle Credential Provider](#).

## Vorrang der Einstellungen

Die Einstellungen für diese Funktion können gleichzeitig verwendet werden, pro Dienst hat jedoch nur ein Wert Priorität. Bei API-Aufrufen an einen bestimmten AWS-Service Wert wird die folgende Reihenfolge verwendet:

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 bei der Instanziierung eines AWS-Service Client- oder Konfigurationsobjekts festlegen.
2. Der Wert, der von einer dienstspezifischen Umgebungsvariablen bereitgestellt wird, z. B. `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-Endpoint-URL für die jeweilige AWS-Service Datei wird zuletzt verwendet.

## Kompatibilität mit AWS SDKs

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

SDK	Unterstützt	Hinweise oder weitere Informationen
<a href="#">AWS CLI v2</a>	Ja	
<a href="#">SDK for C++</a>	Nein	
<a href="#">SDK for Go V2 (1.x)</a>	Ja	
<a href="#">SDK for Go 1.x (V1)</a>	Nein	
<a href="#">SDK for Java 2.x</a>	Nein	
<a href="#">SDK for Java 1.x</a>	Nein	
<a href="#">SDK für 3.x JavaScript</a>	Ja	
<a href="#">SDK für 2.x JavaScript</a>	Nein	
<a href="#">SDK für Kotlin</a>	Ja	
<a href="#">SDK for .NET 3.x</a>	Ja	
<a href="#">SDK for PHP 3.x</a>	Ja	
<a href="#">SDK for Python (Boto3)</a>	Ja	
<a href="#">SDK for Ruby 3.x</a>	Ja	
<a href="#">SDK für Rust</a>	Nein	
<a href="#">Tools für PowerShell</a>	Ja	

## Identifikatoren für dienstspezifische Endpunkte

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

<b>serviceId</b>	Di	<b>AWS_ENDPOINT_URL_&lt;SERVICE&gt;</b>	Umgebungsvariable
	nt		
	üs		
	fü		
	di		
	ge		
	ge		
	Di		
	Al		
	co		
AccessAnalyzer	ac	AWS_ENDPOINT_URL_ACCESSANALYZER	
Account	ac	AWS_ENDPOINT_URL_ACCOUNT	
ACM	ac	AWS_ENDPOINT_URL_ACM	
ACM PCA	ac	AWS_ENDPOINT_URL_ACM_PCA	
Alexa For Business	af	AWS_ENDPOINT_URL_ALEXA_FOR_BUSINESS	
	_l		
amp	ar	AWS_ENDPOINT_URL_AMP	
Amplify	ar	AWS_ENDPOINT_URL_AMPLIFY	
AmplifyBackend	ar	AWS_ENDPOINT_URL_AMPLIFYBACKEND	
	cl		
AmplifyUIBuilder	ar	AWS_ENDPOINT_URL_AMPLIFYUIBUILDER	
	bt		

<b>serviceId</b>	Default endpoint URL	Umgebungsvariable
API Gateway	aws://api-gateway	AWS_ENDPOINT_URL_API_GATEWAY
ApiGatewayManagemementApi	aws://apigatewaymanagementapi	AWS_ENDPOINT_URL_APIGATEWAYMANAGEMENTAPI
ApiGatewayV2	aws://apigatewayv2	AWS_ENDPOINT_URL_APIGATEWAYV2
AppConfig	aws://appconfig	AWS_ENDPOINT_URL_APPCONFIG
AppConfigData	aws://appconfigdata	AWS_ENDPOINT_URL_APPCONFIGDATA
AppFabric	aws://appfabric	AWS_ENDPOINT_URL_APPFABRIC
Appflow	aws://appflow	AWS_ENDPOINT_URL_APPFLOW
AppIntegrations	aws://appintegrations	AWS_ENDPOINT_URL_APPINTEGRATIONS
Application Auto Scaling	aws://application-autoscaling	AWS_ENDPOINT_URL_APPLICATION_AUTO_SCALING

<b>serviceId</b>	<b>D:</b>	<b>AWS_ENDPOINT_URL_&lt;SERVICE&gt;</b>	<b>Umgebungsvariable</b>
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	



<b>serviceId</b>	<b>D:</b>	<b>AWS_ENDPOINT_URL_&lt;SERVICE&gt;</b>	<b>Umgebungsvariable</b>
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	

<b>serviceId</b>	Default endpoint URL	Umgebungsvariable
Bedrock Agent Runtime	<code>aws-bedrock-agent-runtime</code>	<code>AWS_ENDPOINT_URL_BEDROCK_AGENT_RUNTIME</code>
Bedrock Runtime	<code>aws-bedrock</code>	<code>AWS_ENDPOINT_URL_BEDROCK_RUNTIME</code>
billingconductor	<code>aws-billingconductor</code>	<code>AWS_ENDPOINT_URL_BILLINGCONDUCTOR</code>
Braket	<code>aws-braket</code>	<code>AWS_ENDPOINT_URL_BRAKET</code>
Budgets	<code>aws-budgets</code>	<code>AWS_ENDPOINT_URL_BUDGETS</code>
Cost Explorer	<code>aws-costexplorer</code>	<code>AWS_ENDPOINT_URL_COST_EXPLORER</code>
chatbot	<code>aws-chatbot</code>	<code>AWS_ENDPOINT_URL_CHATBOT</code>
Chime	<code>aws-chime</code>	<code>AWS_ENDPOINT_URL_CHIME</code>
Chime SDK Identity	<code>aws-chime-sdk-identity</code>	<code>AWS_ENDPOINT_URL_CHIME_SDK_IDENTITY</code>
Chime SDK Media Pipelines	<code>aws-chime-sdk-media-pipelines</code>	<code>AWS_ENDPOINT_URL_CHIME_SDK_MEDIA_PIPELINES</code>

<b>serviceId</b>	Default endpoint URL	Umgebungsvariable
Chime SDK Meetings	<code>cli.amazonaws.com</code>	<code>AWS_ENDPOINT_URL_CHIME_SDK_MEETINGS</code>
Chime SDK Messaging	<code>cli.amazonaws.com</code>	<code>AWS_ENDPOINT_URL_CHIME_SDK_MESSAGING</code>
Chime SDK Voice	<code>cli.amazonaws.com</code>	<code>AWS_ENDPOINT_URL_CHIME_SDK_VOICE</code>
CleanRooms	<code>cli.amazonaws.com</code>	<code>AWS_ENDPOINT_URL_CLEANROOMS</code>
CleanRoomsML	<code>cli.amazonaws.com</code>	<code>AWS_ENDPOINT_URL_CLEANROOMSML</code>
Cloud9	<code>cli.amazonaws.com</code>	<code>AWS_ENDPOINT_URL_CLOUD9</code>
CloudControl	<code>cli.amazonaws.com</code>	<code>AWS_ENDPOINT_URL_CLOUDCONTROL</code>
CloudDirectory	<code>cli.amazonaws.com</code>	<code>AWS_ENDPOINT_URL_CLOUDDIRECTORY</code>
CloudFormation	<code>cli.amazonaws.com</code>	<code>AWS_ENDPOINT_URL_CLOUDFORMATION</code>

<b>serviceId</b>	Default endpoint URL	Umgebungsvariable
CloudFront	c: AWS_ENDPOINT_URL_CLOUDFRONT	
CloudFront Key Value Store	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	

<b>serviceId</b>	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>

<b>serviceId</b>	Definition	Umweltvariable
	Die Umgebungsvariable <code>AWS_ENDPOINT_URL_&lt;SERVICE&gt;</code> wird für die Kommunikation mit dem Endpunkt für den Service <code>&lt;SERVICE&gt;</code> verwendet.	Umgebungsvariable
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>	

<b>serviceId</b>	<b>D</b>	<b>AWS_ENDPOINT_URL_&lt;SERVICE&gt;</b>	<b>Umgebungsvariable</b>
Config Service	co	AWS_ENDPOINT_URL_CONFIG_SERVICE	
Connect	co	AWS_ENDPOINT_URL_CONNECT	
Connect Contact Lens	co	AWS_ENDPOINT_URL_CONNECT_CONTACT_LENS	
ConnectCampaigns	co	AWS_ENDPOINT_URL_CONNECTCAMPAIGNS	
ConnectCases	co	AWS_ENDPOINT_URL_CONNECTCASES	
ConnectParticipant	co	AWS_ENDPOINT_URL_CONNECTPARTICIPANT	
ControlTower	co	AWS_ENDPOINT_URL_CONTROLTOWER	
Cost Optimization Hub	co	AWS_ENDPOINT_URL_COST_OPTIMIZATION_HUB	

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



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

<b>serviceId</b>	Default endpoint URL	Environment variable
	<code>AWS_ENDPOINT_URL_&lt;SERVICE&gt;</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>	

<b>serviceId</b>	Di nu üs fü di ge ge D: A: c:	<b>AWS_ENDPOINT_URL_&lt;SERVICE&gt;</b> Umgebungsvariable
ElastiCache	e: h:	AWS_ENDPOINT_URL_ELASTICACHE
Elastic Beanstalk	e: e:	AWS_ENDPOINT_URL_ELASTIC_BEANSTALK
Elastic Transcoder	e: r:	AWS_ENDPOINT_URL_ELASTIC_TRANSCODER
Elastic Load Balancing	e: o: c:	AWS_ENDPOINT_URL_ELASTIC_LOAD_BALANCING
Elastic Load Balancing v2	e: o: c:	AWS_ENDPOINT_URL_ELASTIC_LOAD_BALANCING_V2
EMR	e: r:	AWS_ENDPOINT_URL_EMR
EMR containers	e: i:	AWS_ENDPOINT_URL_EMR_CONTAINERS
EMR Serverless	e: r:	AWS_ENDPOINT_URL_EMR_SERVERLESS

<b>serviceId</b>	Die Umgebungsvariable <b>AWS_ENDPOINT_URL_&lt;SERVICE&gt;</b> wird für die Generierung der API-Client-URL verwendet.
EntityResolution	Environment Variable: <b>AWS_ENDPOINT_URL_ENTITYRESOLUTION</b>
Elasticsearch Service	Environment Variable: <b>AWS_ENDPOINT_URL_ELASTICSEARCH_SERVICE</b>
EventBridge	Environment Variable: <b>AWS_ENDPOINT_URL_EVENTBRIDGE</b>
Evidently	Environment Variable: <b>AWS_ENDPOINT_URL_EVIDENTLY</b>
finspace	Environment Variable: <b>AWS_ENDPOINT_URL_FINSPLACE</b>
finspace data	Environment Variable: <b>AWS_ENDPOINT_URL_FINSPLACE_DATA</b>
Firehose	Environment Variable: <b>AWS_ENDPOINT_URL_FIREHOSE</b>
fis	Environment Variable: <b>AWS_ENDPOINT_URL_FIS</b>
FMS	Environment Variable: <b>AWS_ENDPOINT_URL_FMS</b>
forecast	Environment Variable: <b>AWS_ENDPOINT_URL_FORECAST</b>
forecastquery	Environment Variable: <b>AWS_ENDPOINT_URL_FORECASTQUERY</b>

<b>serviceId</b>	<b>D:</b> <b>AWS_ENDPOINT_URL_&lt;SERVICE&gt;</b> <b>Umgebungsvariable</b>
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:

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

<b>serviceId</b>	<b>D</b>	<b>AWS_ENDPOINT_URL_&lt;SERVICE&gt;</b>	<b>Umgebungsvariable</b>
Inspector2	i	AWS_ENDPOINT_URL_INSPECTOR2	
InternetMonitor	i	AWS_ENDPOINT_URL_INTERNETMONITOR	
IoT	i	AWS_ENDPOINT_URL_IOT	
IoT Data Plane	i	AWS_ENDPOINT_URL_IOT_DATA_PLANE	
IoT Jobs Data Plane	i	AWS_ENDPOINT_URL_IOT_JOBS_DATA_PLANE	
IoT 1Click Devices Service	i	AWS_ENDPOINT_URL_IOT_1CLICK_DEVICES_SERVICE	
IoT 1Click Projects	i	AWS_ENDPOINT_URL_IOT_1CLICK_PROJECTS	
IoTAnalytics	i	AWS_ENDPOINT_URL_IOTANALYTICS	

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



<b>serviceId</b>	<b>Default endpoint URL</b>	<b>Umgebungsvariable</b>
	<code>AWS_ENDPOINT_URL_&lt;SERVICE&gt;</code>	
IoT Wireless	<code>AWS_ENDPOINT_URL_IOT_WIRELESS</code>	
ivs	<code>AWS_ENDPOINT_URL_IVS</code>	
IVS RealTime	<code>AWS_ENDPOINT_URL_IVS_REALTIME</code>	
ivschat	<code>AWS_ENDPOINT_URL_IVSCHAT</code>	
Kafka	<code>AWS_ENDPOINT_URL_KAFKA</code>	
KafkaConnect	<code>AWS_ENDPOINT_URL_KAFKACONNECT</code>	
kendra	<code>AWS_ENDPOINT_URL_KENDRA</code>	
Kendra Ranking	<code>AWS_ENDPOINT_URL_KENDRA_RANKING</code>	
Keyspaces	<code>AWS_ENDPOINT_URL_KEYSPACES</code>	
Kinesis	<code>AWS_ENDPOINT_URL_KINESIS</code>	

<b>serviceId</b>	<b>D:</b> <b>AWS_ENDPOINT_URL_&lt;SERVICE&gt;</b> 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

<b>serviceId</b>	<b>D:</b> <b>AWS_ENDPOINT_URL_&lt;SERVICE&gt;</b> 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:

<b>serviceId</b>	<b>D:</b> <b>AWS_ENDPOINT_URL_&lt;SERVICE&gt;</b> Umgebungsvariable
License Manager Linux Subscriptions	<b>l:</b> AWS_ENDPOINT_URL_LICENSE_MANAGER_LINUX_SUBSCRIPTIONS
License Manager User Subscriptions	<b>l:</b> AWS_ENDPOINT_URL_LICENSE_MANAGER_USER_SUBSCRIPTIONS
Lightsail	<b>l:</b> AWS_ENDPOINT_URL_LIGHTSAIL
Location	<b>l:</b> AWS_ENDPOINT_URL_LOCATION
CloudWatch Logs	<b>c:</b> AWS_ENDPOINT_URL_CLOUDWATCH_LOGS
CloudWatch Logs	<b>c:</b> AWS_ENDPOINT_URL_CLOUDWATCH_LOGS
LookoutEquipment	<b>l:</b> AWS_ENDPOINT_URL_LOOKOUTEQUIPMENT
LookoutMetrics	<b>l:</b> AWS_ENDPOINT_URL_LOOKOUTMETRICS

<b>serviceId</b>	Definition	Umgebungsvariable
LookoutVision	Die Lookout Vision API verwendet die Endpunkte <code>lookoutvision.amazonaws.com</code> und <code>lookoutvision.amazonaws.com.cn</code> für die Region <code>us-east-1</code> . Die Lookout Vision API verwendet die Endpunkte <code>lookoutvision.amazonaws.com</code> und <code>lookoutvision.amazonaws.com.cn</code> für die Region <code>us-east-1</code> .	<code>AWS_ENDPOINT_URL_LOOKOUTVISION</code>
m2	Die Amazon Macie 2 API verwendet die Endpunkte <code>macie2.amazonaws.com</code> und <code>macie2.amazonaws.com.cn</code> für die Region <code>us-east-1</code> .	<code>AWS_ENDPOINT_URL_M2</code>
Machine Learning	Die Amazon Machine Learning API verwendet die Endpunkte <code>ml.amazonaws.com</code> und <code>ml.amazonaws.com.cn</code> für die Region <code>us-east-1</code> .	<code>AWS_ENDPOINT_URL_MACHINE_LEARNING</code>
Macie2	Die Amazon Macie 2 API verwendet die Endpunkte <code>macie2.amazonaws.com</code> und <code>macie2.amazonaws.com.cn</code> für die Region <code>us-east-1</code> .	<code>AWS_ENDPOINT_URL_MACIE2</code>
ManagedBlockchain	Die Amazon Managed Blockchain API verwendet die Endpunkte <code>managedblockchain.amazonaws.com</code> und <code>managedblockchain.amazonaws.com.cn</code> für die Region <code>us-east-1</code> .	<code>AWS_ENDPOINT_URL_MANAGEDBLOCKCHAIN</code>
ManagedBlockchain Query	Die Amazon Managed Blockchain Query API verwendet die Endpunkte <code>managedblockchain-query.amazonaws.com</code> und <code>managedblockchain-query.amazonaws.com.cn</code> für die Region <code>us-east-1</code> .	<code>AWS_ENDPOINT_URL_MANAGEDBLOCKCHAIN_QUERY</code>
Marketplace Agreement	Die Amazon Marketplace Agreement API verwendet die Endpunkte <code>marketplace-agreement.amazonaws.com</code> und <code>marketplace-agreement.amazonaws.com.cn</code> für die Region <code>us-east-1</code> .	<code>AWS_ENDPOINT_URL_MARKETPLACE_AGREEMENT</code>
Marketplace Catalog	Die Amazon Marketplace Catalog API verwendet die Endpunkte <code>marketplace-catalog.amazonaws.com</code> und <code>marketplace-catalog.amazonaws.com.cn</code> für die Region <code>us-east-1</code> .	<code>AWS_ENDPOINT_URL_MARKETPLACE_CATALOG</code>

<b>serviceId</b>	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

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

<b>serviceId</b>	D: <b>AWS_ENDPOINT_URL_&lt;SERVICE&gt;</b> 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



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

<b>serviceId</b>	Default endpoint URL	Umgebungsvariable
	<code>aws://&lt;SERVICE&gt;</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>

<b>serviceId</b>	Default endpoint URL	Umgebungsvariable
Personalize	personalize.amazonaws.com	AWS_ENDPOINT_URL_PERSONALIZE
Personalize Events	personalize.amazonaws.com	AWS_ENDPOINT_URL_PERSONALIZE_EVENTS
Personalize Runtime	personalize.amazonaws.com	AWS_ENDPOINT_URL_PERSONALIZE_RUNTIME
PI	personalize.amazonaws.com	AWS_ENDPOINT_URL_PI
Pinpoint	pinpoint.amazonaws.com	AWS_ENDPOINT_URL_PINPOINT
Pinpoint Email	pinpoint.amazonaws.com	AWS_ENDPOINT_URL_PINPOINT_EMAIL
Pinpoint SMS Voice	pinpoint.amazonaws.com	AWS_ENDPOINT_URL_PINPOINT_SMS_VOICE
Pinpoint SMS Voice V2	pinpoint.amazonaws.com	AWS_ENDPOINT_URL_PINPOINT_SMS_VOICE_V2
Pipes	pipes.amazonaws.com	AWS_ENDPOINT_URL_PIPES

<b>serviceId</b>	Default endpoint URL	Umgebungsvariable
	<code>DefaultEndpointURL</code>	
Polly	<code>pol</code>	<code>AWS_ENDPOINT_URL_POLLY</code>
Pricing	<code>p</code>	<code>AWS_ENDPOINT_URL_PRICING</code>
PrivateNetworks	<code>pn</code>	<code>AWS_ENDPOINT_URL_PRIVATENETWORKS</code>
Proton	<code>p</code>	<code>AWS_ENDPOINT_URL_PROTON</code>
QBusiness	<code>qb</code>	<code>AWS_ENDPOINT_URL_QBUSINESS</code>
QConnect	<code>qc</code>	<code>AWS_ENDPOINT_URL_QCONNECT</code>
QLDB	<code>q</code>	<code>AWS_ENDPOINT_URL_QLDB</code>
QLDB Session	<code>qs</code>	<code>AWS_ENDPOINT_URL_QLDB_SESSION</code>
QuickSight	<code>qs</code>	<code>AWS_ENDPOINT_URL_QUICKSIGHT</code>
RAM	<code>r</code>	<code>AWS_ENDPOINT_URL_RAM</code>
rbin	<code>rb</code>	<code>AWS_ENDPOINT_URL_RBIN</code>
RDS	<code>r</code>	<code>AWS_ENDPOINT_URL_RDS</code>
RDS Data	<code>rd</code>	<code>AWS_ENDPOINT_URL_RDS_DATA</code>

<b>serviceId</b>	Di	<b>AWS_ENDPOINT_URL_&lt;SERVICE&gt;</b>	Umgebungsvariable
Redshift	re	AWS_ENDPOINT_URL_REDSHIFT	
Redshift Data	re	AWS_ENDPOINT_URL_REDSHIFT_DATA	
Redshift Serverless	re	AWS_ENDPOINT_URL_REDSHIFT_SERVERLESS	
Rekognition	re	AWS_ENDPOINT_URL_REKOGNITION	
repostspace	re	AWS_ENDPOINT_URL_REPOSTSPACE	
resiliencehub	re	AWS_ENDPOINT_URL_RESILIENCEHUB	
Resource Explorer 2	re	AWS_ENDPOINT_URL_RESOURCE_EXPLORER_2	
Resource Groups	re	AWS_ENDPOINT_URL_RESOURCE_GROUPS	

<b>serviceId</b>	Default endpoint URL	Environment variable
Resource Groups Tagging API	<code>AWS_ENDPOINT_URL_RESOURCE_GROUPS_TAGGING_API</code>	
RoboMaker	<code>AWS_ENDPOINT_URL_ROBOMAKER</code>	
RolesAnywhere	<code>AWS_ENDPOINT_URL_ROLESANYPWHERE</code>	
Route 53	<code>AWS_ENDPOINT_URL_ROUTE_53</code>	
Route53 Recovery Cluster	<code>AWS_ENDPOINT_URL_ROUTE53_RECOVERY_CLUSTER_ENDPOINT</code>	
Route53 Recovery Control Config	<code>AWS_ENDPOINT_URL_ROUTE53_RECOVERY_CONTROL_CONFIG</code>	
Route53 Recovery Readiness	<code>AWS_ENDPOINT_URL_ROUTE53_RECOVERY_READINESS_ENDPOINT</code>	

<b>serviceId</b>	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>	

<b>serviceId</b>	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>	



<b>serviceId</b>	Default endpoint URL	Umgebungsvariable
SecurityHub	<code>aws://logs-<i>us-east-1</i>-<i>awslogs</i>-<i>us-east-1</i>.amazonaws.com</code>	<code>AWS_ENDPOINT_URL_SECURITYHUB</code>
SecurityLake	<code>aws://logs-<i>us-east-1</i>-<i>awslogs</i>-<i>us-east-1</i>.amazonaws.com</code>	<code>AWS_ENDPOINT_URL_SECURITYLAKE</code>
ServerlessApplicationRepository	<code>aws://logs-<i>us-east-1</i>-<i>awslogs</i>-<i>us-east-1</i>.amazonaws.com</code>	<code>AWS_ENDPOINT_URL_SERVERLESSAPPLICATIONREPOSITORY</code>
Service Quotas	<code>aws://logs-<i>us-east-1</i>-<i>awslogs</i>-<i>us-east-1</i>.amazonaws.com</code>	<code>AWS_ENDPOINT_URL_SERVICE_QUOTAS</code>
Service Catalog	<code>aws://logs-<i>us-east-1</i>-<i>awslogs</i>-<i>us-east-1</i>.amazonaws.com</code>	<code>AWS_ENDPOINT_URL_SERVICE_CATALOG</code>
Service Catalog AppRegistry	<code>aws://logs-<i>us-east-1</i>-<i>awslogs</i>-<i>us-east-1</i>.amazonaws.com</code>	<code>AWS_ENDPOINT_URL_SERVICE_CATALOG_APP_REGISTRY</code>
ServiceDiscovery	<code>aws://logs-<i>us-east-1</i>-<i>awslogs</i>-<i>us-east-1</i>.amazonaws.com</code>	<code>AWS_ENDPOINT_URL_SERVICEDISCOVERY</code>
SES	<code>aws://logs-<i>us-east-1</i>-<i>awslogs</i>-<i>us-east-1</i>.amazonaws.com</code>	<code>AWS_ENDPOINT_URL_SES</code>

<b>serviceId</b>	<b>D:</b> <b>AWS_ENDPOINT_URL_&lt;SERVICE&gt;</b>	Umgebungsvariable
SESV2	s: AWS_ENDPOINT_URL_SESV2	
Shield	s: AWS_ENDPOINT_URL_SHIELD	
signer	s: AWS_ENDPOINT_URL_SIGNER	
SimSpaceWeaver	s: AWS_ENDPOINT_URL_SIMSPACEWEAVER	
SMS	s: AWS_ENDPOINT_URL_SMS	
Snow Device Management	s: AWS_ENDPOINT_URL_SNOW_DEVICE_MANAGEMENT	
Snowball	s: AWS_ENDPOINT_URL_SNOWBALL	
SNS	s: AWS_ENDPOINT_URL_SNS	
SQS	s: AWS_ENDPOINT_URL_SQS	
SSM	s: AWS_ENDPOINT_URL_SSM	
SSM Contacts	s: AWS_ENDPOINT_URL_SSM_CONTACTS	
SSM Incidents	s: AWS_ENDPOINT_URL_SSM_INCIDENTS	

<b>serviceId</b>	<b>D:</b> <b>AWS_ENDPOINT_URL_&lt;SERVICE&gt;</b>	<b>Umgebungsvariable</b>
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	

<b>serviceId</b>	<b>D:</b> <b>AWS_ENDPOINT_URL_&lt;SERVICE&gt;</b> 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

<b>serviceId</b>	<b>D</b>	<b>AWS_ENDPOINT_URL_&lt;SERVICE&gt;</b>	<b>Umgebungsvariable</b>
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	

<b>serviceId</b>	D: <b>AWS_ENDPOINT_URL_&lt;SERVICE&gt;</b> n üs fü di ge ge D: A/ c	Umgebungsvariable
WorkSpaces	w: <b>AWS_ENDPOINT_URL_WORKSPACES</b> s	
WorkSpaces Thin Client	w: <b>AWS_ENDPOINT_URL_WORKSPACES_THIN_CLIENT</b> s_ i	
WorkSpaces Web	w: <b>AWS_ENDPOINT_URL_WORKSPACES_WEB</b> s_	
XRay	x: <b>AWS_ENDPOINT_URL_XRAY</b>	

## Standardeinstellungen für intelligente Konfigurationen

Mit der Funktion für intelligente Konfigurationsstandardwerte können AWS SDKs vordefinierte, optimierte Standardwerte für andere Konfigurationseinstellungen bereitstellen.

Konfigurieren Sie diese Funktionalität wie folgt:

**defaults\_mode**- Einstellung für gemeinsam genutzte AWS **config** Dateien,

**AWS\_DEFAULTS\_MODE**- Umgebungsvariable, **aws.defaultsMode**- JVM-Systemeigenschaft: Nur Java/Kotlin

Mit dieser Einstellung können Sie einen Modus wählen, der zu Ihrer Anwendungsarchitektur passt und dann optimierte Standardwerte für Ihre Anwendung bereitstellt. Wenn für eine AWS SDK-Einstellung ein Wert explizit festgelegt ist, hat dieser Wert immer Vorrang. Wenn für eine AWS SDK-Einstellung kein explizit festgelegter Wert festgelegt wurde und sie nicht dem

`defaults_mode` Wert einer Legacy-Einstellung entspricht, kann diese Funktion unterschiedliche Standardwerte für verschiedene Einstellungen bereitstellen, die für Ihre Anwendung optimiert sind. Zu den Einstellungen können Folgendes gehören: HTTP-Kommunikationseinstellungen, Wiederholungsverhalten, regionale Endpunkteinstellungen des Dienstes und möglicherweise jede SDK-bezogene Konfiguration. Kunden, die diese Funktion verwenden, können neue Standardkonfigurationen erhalten, die auf allgemeine Nutzungsszenarien zugeschnitten sind. Wenn Ihre nicht identisch `defaults_mode` ist, empfehlen wir `legacy`, Tests Ihrer Anwendung durchzuführen, wenn Sie das SDK aktualisieren, da sich die angegebenen Standardwerte ändern können, wenn sich die bewährten Methoden weiterentwickeln.

Standardwert: `legacy`

Hinweis: Neue Hauptversionen von SDKs werden standardmäßig verwendet. `standard`

Zulässige Werte:

- `legacy`— Stellt Standardeinstellungen bereit, die je nach SDK variieren und vor der Einrichtung von `defaults_mode` existierten.
- `standard`— Stellt die neuesten empfohlenen Standardwerte bereit, deren Ausführung in den meisten Szenarien sicher sein sollte.
- `in-region`— Baut auf dem Standardmodus auf und beinhaltet eine Optimierung, die auf Anwendungen zugeschnitten ist, die AWS-Services aus demselben Modus heraus aufrufen AWS-Region.
- `cross-region`— Baut auf dem Standardmodus auf und beinhaltet eine Optimierung, die auf Anwendungen zugeschnitten ist, die AWS-Services in einer anderen Region aufrufen.
- `mobile`— Baut auf dem Standardmodus auf und beinhaltet eine auf mobile Anwendungen zugeschnittene Optimierung.
- `auto`— Baut auf dem Standardmodus auf und beinhaltet experimentelle Funktionen. Das SDK versucht, die Laufzeitumgebung zu ermitteln, um die entsprechenden Einstellungen automatisch zu ermitteln. Die `auto` Erkennung basiert auf Heuristik und bietet keine hundertprozentige Genauigkeit. Wenn die Laufzeitumgebung nicht bestimmt werden kann, `standard` wird der Modus verwendet. Die `auto` Erkennung fragt möglicherweise [Instanzmetadaten und Benutzerdaten 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 optimiert werdendefaults\_mode:

- `retryMode`— Gibt an, wie das SDK versucht, es erneut zu versuchen. Siehe [Wiederholungsverhalten](#).
- `stsRegionalEndpoints`— Gibt an, wie das SDK den AWS-Service Endpunkt bestimmt, über den es mit AWS Security Token Service (AWS STS) kommuniziert. Siehe [AWS STS Regionalisierte Endpunkte](#).
- `s3UsEast1RegionalEndpoints`— Gibt an, wie das SDK den AWS Service-Endpunkt bestimmt, den es für die Kommunikation mit Amazon S3 für die us-east-1 Region verwendet.
- `connectTimeoutInMillis`— Nach einem ersten Verbindungsversuch auf einem Socket, die Zeitspanne bis zum Timeout. Wenn der Client den Abschluss des Connect-Handshakes nicht erhält, gibt der Client auf und schlägt den Vorgang fehl.
- `tlsNegotiationTimeoutInMillis`— Die maximale Zeit, die ein TLS-Handshake vom Senden der CLIENT HELLO-Nachricht bis zu dem Zeitpunkt in Anspruch nehmen kann, zu dem der Client und der Server die Chiffren vollständig ausgehandelt und Schlüssel ausgetauscht haben.

Der Standardwert für jede Einstellung ändert sich je nach den Einstellungen, die Sie für Ihre Anwendung defaults\_mode ausgewählt haben. Diese Werte sind derzeit wie folgt festgelegt (Änderungen vorbehalten):

Parameter	Modus <b>standard</b>	Modus <b>in-region</b>	Modus <b>cross-region</b>	Modus <b>mobile</b>
<code>retryMode</code>	standard	standard	standard	standard
<code>stsRegionalEndpoints</code>	regional	regional	regional	regional



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

Wenn `defaults_mode` Sie beispielsweise „“ ausgewählt haben `standard`, wird der `standard` Wert für `retry_mode` (aus den gültigen `retry_mode` Optionen) und der `regional` Wert für `stsRegionalEndpoints` (aus den gültigen `stsRegionalEndpoints` Optionen) zugewiesen.

## Kompatibilität mit AWS SDKs

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

SDK	Unterstützt	Hinweise oder weitere Informationen
<a href="#">AWS CLI v2</a>	Nein	
<a href="#">SDK for C++</a>	Ja	Parameter sind nicht optimiert <code>:stsRegionalEndpoints</code> , <code>s3UsEast1RegionalEndpoints</code> , <code>tlsNegoti</code>

SDK	Unterstützt	Hinweise oder weitere Informationen
		<code>ationTimeoutInMillis</code> .
<a href="#">SDK for Go V2 (1.x)</a>	Ja	Parameter nicht optimiert <code>:retryMode</code> , <code>stsRegionalEndpoints</code> , <code>s3UsEast1RegionalEndpoints</code> .
<a href="#">SDK for Go 1.x (V1)</a>	Nein	
<a href="#">SDK for Java 2.x</a>	Ja	Parameter sind nicht optimiert <code>:stsRegionalEndpoints</code> .
<a href="#">SDK for Java 1.x</a>	Nein	
<a href="#">SDK für 3.x JavaScript</a>	Ja	Parameter nicht optimiert <code>:stsRegionalEndpoints</code> , <code>s3UsEast1RegionalEndpoints</code> , <code>tlsNegotiationTimeoutInMillis</code> . <code>connectTimeoutInMillis</code> heißt <code>connectionTimeout</code> .
<a href="#">SDK für JavaScript 2.x</a>	Nein	
<a href="#">SDK für Kotlin</a>	Nein	
<a href="#">SDK for .NET 3.x</a>	Ja	Parameter nicht optimiert <code>:connectTimeoutInMillis</code> , <code>tlsNegotiationTimeoutInMillis</code> .

SDK	Unterstützt	Hinweise oder weitere Informationen
<a href="#">SDK for PHP 3.x</a>	Ja	Parameter nicht optimiert :tlsNegotiationTime outInMillis .
<a href="#">SDK for Python (Boto3)</a>	Ja	Parameter nicht optimiert: tlsNegotiationTime outInMillis
<a href="#">SDK for Ruby 3.x</a>	Ja	
<a href="#">SDK für Rust</a>	Nein	
<a href="#">Tools für PowerShell</a>	Ja	Parameter nicht optimiert :connectTimeoutInMi llis ,tlsNegoti ationTimeoutInMill is .

# AWSCommon Runtime (CRT) -Bibliotheken

Die AWS Common Runtime (CRT) -Bibliotheken sind eine Basisbibliothek der SDKs. CRT ist eine modulare Familie unabhängiger Pakete, die in C geschrieben sind. Jedes Paket bietet eine gute Leistung und minimalen Platzbedarf für verschiedene erforderliche Funktionen. Diese Funktionen sind allen SDKs gemeinsam und sorgen für eine bessere Wiederverwendung, Optimierung und Genauigkeit von Code. Die Pakete sind:

- [awslabs/aws-c-auth](#): AWS clientseitige Authentifizierung (Standardanbieter für Anmeldeinformationen und Signierung (sigv4))
- [awslabs/aws-c-cal](#): Primitive kryptografische Typen, Hashes (MD5, SHA256, SHA256 HMAC), Unterzeichner, AES
- [awslabs/aws-c-common](#): Grundlegende Datenstrukturen, primitive Typen für Threading/Synchronisation, 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 Remoteprozeduraufrufen (RPC) über Event-Streams
- [awslabs/aws-c-http](#): C99-Implementierung der HTTP/1.1- und HTTP/2-Spezifikationen
- [awslabs/aws-c-io](#): Sockets (TCP, UDP), DNS, Pipes, Ereignisschleifen, Kanäle, SSL/TLS
- [awslabs/aws-c-iot](#): C99-Implementierung der Integration von AWS IoT-Cloud-Diensten mit Geräten
- [awslabs/aws-c-mqtt](#): Standardmäßiges, leichtes Messaging-Protokoll für das Internet der Dinge (IoT)
- [awslabs/aws-c-s3](#): C99-Bibliotheksimplementierung für die Kommunikation mit dem Amazon S3 S3-Service, konzipiert für die Maximierung des Durchsatzes auf Amazon EC2 EC2-Instances mit hoher Bandbreite
- [awslabs/aws-c-sdkutils](#): Eine Dienstprogramm-Bibliothek zum Analysieren und Verwalten von Profilen AWS
- [awslabs/aws-checksums](#): Plattformübergreifende hardwarebeschleunigte CRC32c und CRC32 mit Rückgriff auf effiziente Softwareimplementierungen
- [awslabs/aws-1c](#): Kryptografische Allzweckbibliothek, die vom AWS Cryptography-Team AWS und seinen Kunden verwaltet wird und auf Code aus dem Google BoringSSL-Projekt und dem OpenSSL-Projekt basiert

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

Das CRT ist über alle SDKs außer Go verfügbar.

## CRT-Abhängigkeiten

Die CRT-Bibliotheken bilden ein komplexes Netz von Beziehungen und Abhängigkeiten. Die Kenntnis dieser Beziehungen ist hilfreich, wenn Sie das CRT direkt aus dem Quellcode erstellen müssen.

Die meisten Benutzer greifen jedoch über ihr Sprach-SDK (wie SDK for C++ oder AWS SDK for Java) oder ihr Sprach-IoT-Geräte-SDK (wie IoT SDK for C++ oder AWS IoT SDK for Java) auf CRT-Funktionen zu. In der folgenden Abbildung bezieht sich das Feld Sprach-CRT-Bindungen auf das Paket, das die CRT-Bibliotheken für ein bestimmtes Sprach-SDK umschließt. Dies ist eine Sammlung von Paketen in der Form `aws-crt-*`, wobei `*` für eine SDK-Sprache steht (z. B. [aws-crt-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. [Matrix zur Versionsunterstützung](#)

# AWS Versionsunterstützungsmatrix für SDKs und Tools

Die folgende Matrix zeigt die Liste der verfügbaren Hauptversionen des AWS Software Development Kit (SDK) und zeigt, wo sie sich im Wartungslebenszyklus befinden, sowie die zugehörigen Zeitpläne. Ausführliche Informationen zum Lebenszyklus der Hauptversionen von AWS SDKs und Tools und den zugrunde liegenden Abhängigkeiten finden Sie unter [Wartungsrichtlinie](#)

SDK	Hauptversion	Aktuelle Phase	Datum der allgemeinen Verfügbarkeit	Hinweise
<a href="#">AWS CLI</a>	1.x	Allgemeine Verfügbarkeit	9/2/2013	
<a href="#">AWS CLI</a>	2.x	Allgemeine Verfügbarkeit	10.02.2020	
<a href="#">SDK for C++</a>	1.x	Allgemeine Verfügbarkeit	9/2/2015	
<a href="#">SDK for Go V2</a>	V2 1.x	Allgemeine Verfügbarkeit	19.1.2021	
<a href="#">SDK for Go</a>	1.x	Ankündigung von Wartungsarbeiten	19.11.2015	Einzelheiten und Termine finden Sie in der <a href="#">Ankündigung</a>
<a href="#">SDK für Java</a>	1.x	Ankündigung von Wartungsarbeiten	25.03.2010	Einzelheiten und Termine finden Sie in der <a href="#">Ankündigung</a>
<a href="#">SDK für Java</a>	2.x	Allgemeine Verfügbarkeit	20.11.2018	
<a href="#">SDK für JavaScript</a>	1.x	Ende des Supports	06.05.2013	

SDK	Hauptversion	Aktuelle Phase	Datum der allgemeinen Verfügbarkeit	Hinweise
<a href="#">SDK für JavaScript</a>	2.x	Ankündigung von Wartungsarbeiten	19.06.2014	Einzelheiten und Termine finden Sie in der <a href="#">Ankündigung</a>
<a href="#">SDK für JavaScript</a>	3.x	Allgemeine Verfügbarkeit	15.12.2020	
<a href="#">SDK für Kotlin</a>	1.x	Allgemeine Verfügbarkeit	27.11.2023	
<a href="#">SDK for .NET</a>	1.x	Ende des Supports	11/2009	
<a href="#">SDK for .NET</a>	2.x	Ende des Supports	8.11.2013	
<a href="#">SDK for .NET</a>	3.x	Allgemeine Verfügbarkeit	28.07.2015	
<a href="#">SDK for PHP</a>	2.x	Ende des Supports	02.11.2012	
<a href="#">SDK for PHP</a>	3.x	Allgemeine Verfügbarkeit	27.5.2015	
<a href="#">SDK für Python (Boto2)</a>	1.x	Ende des Supports	13.07.2011	
<a href="#">SDK for Python (Boto3)</a>	1.x	Allgemeine Verfügbarkeit	22.06.2015	
<a href="#">SDK für Python (Botocore)</a>	1.x	Allgemeine Verfügbarkeit	22.06.2015	

SDK	Hauptversion	Aktuelle Phase	Datum der allgemeinen Verfügbarkeit	Hinweise
<a href="#">SDK for Ruby</a>	1.x	Ende des Supports	14.7.2011	
<a href="#">SDK for Ruby</a>	2.x	Ende des Supports	15.02.2015	
<a href="#">SDK for Ruby</a>	3.x	Allgemeine Verfügbarkeit	29.8.2017	
<a href="#">SDK für Rust</a>	1.x	Allgemeine Verfügbarkeit	27.11.2023	
<a href="#">SDK für Swift</a>	1.x	Developer-Vorschau		
<a href="#">Tools für PowerShell</a>	2.x	Ende des Supports	8.11.2013	
<a href="#">Werkzeuge für PowerShell</a>	3.x	Ende des Supports	29.7.2015	
<a href="#">Werkzeuge für PowerShell</a>	4.x	Allgemeine Verfügbarkeit	21.11.2019	

# Referenzhandbuch zur Dokumentenhistorie für AWS SDKs und Tools

In der folgenden Tabelle werden wichtige Ergänzungen und Aktualisierungen des Referenzhandbuchs für AWS SDKs und Tools beschrieben. Um Benachrichtigungen über Aktualisierungen dieser Dokumentation zu erhalten, können Sie den RSS-Feed abonnieren.

Änderung	Beschreibung	Datum
<a href="#">SDK for Java 1.x-Systemeigenschaften</a>	Fügen Sie Details zu den unterstützten JVM-Systemkonfigurationseinstellungen von Version 1.x hinzu. AWS SDK for Java	30. Mai 2024
<a href="#">Aktualisierungen der Einstellungen</a>	Fügen Sie JVM-Systemkonfigurationseinstellungen hinzu.	27. März 2024
<a href="#">Aktualisierungen der Kompatibilitätstabelle</a>	Aktualisierungen der Kompatibilität für die SDK-Unterstützung, Aktualisierungen der IAM Identity Center-Verfahren.	20. Februar 2024
<a href="#">Aktualisierung der Container-Anmeldeinformationen. IMDS-Aktualisierung.</a>	Unterstützung für Amazon EKS wird hinzugefügt. Einstellung zur Deaktivierung des IMDSv1-Fallbacks hinzugefügt.	29. Dezember 2023
<a href="#">Komprimierung anfordern</a>	Einstellungen für die Funktion zur Komprimierung von Anfragen werden hinzugefügt.	27. Dezember 2023
<a href="#">Kompatibilitätstabellen</a>	Die Kompatibilitätstabellen für SDK- und Toolfunktionen	10. Dezember 2023

wurden aktualisiert und enthalten nun SDK für Kotlin, SDK für Rust und AWS Tools for PowerShell.

### [Aktualisierungen der Authentifizierung](#)

Aktualisierungen der unterstützten Authentifizierungsmethoden für SDKs und Tools.

1. Juli 2023

### [Aktualisierungen der bewährten Methoden für IAM](#)

Aktualisierter Leitfaden, angepasst an die bewährten IAM-Methoden. Weitere Informationen finden Sie unter [Bewährte IAM-Methoden](#).

27. Februar 2023

### [SSO-Aktualisierungen](#)

Aktualisierungen der SSO-Anmeldeinformationen für die neue SSO-Token-Konfiguration.

19. November 2022

### [Aktualisierungen der Einstellungen](#)

Aktualisierungen der Unterstützungstabelle für die allgemeine Konfiguration und für Amazon S3 Multi-Region Access Points.

17. November 2022

### [Aktualisierungen der Einstellungen](#)

Aktualisierungen zur besseren Übersicht der IMDS-Client- und IMDS-Anmeldeinformationen. Aktualisierungen der Umgebungsvariablen.

04. November 2022

### [Die Willkommenseite wird aktualisiert](#)

Ankündigung von Amazon CodeWhisperer.

22. September 2022

---

<a href="#">Änderung des Dienstnamens für Single Sign-On</a>	Aktualisierungen, die dem Umstand Rechnung tragen, dass AWS SSO jetzt als AWS IAM Identity Center bezeichnet wird.	26. Juli 2022
<a href="#">Die Einstellungen werden aktualisiert</a>	Kleinere Aktualisierungen der Details der Konfigurationsdatei und der unterstützten Einstellungen.	15. Juni 2022
<a href="#">Aktualisieren</a>	Umfangreiches Update fast aller Teile dieses Handbuchs.	1. Februar 2022
<a href="#">Erstversion</a>	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.