Guida di riferimento

AWS SDK e strumenti

AWS SDK e strumenti: Guida di riferimento

Table of Contents

AWS Guida di riferimento agli SDK e agli strumenti 1
Risorse per sviluppatori 2
Notifica di telemetria del Toolkit 3
Configurazione 4
Condivisi `config` e `credentials` file 5
Profili 5
Formato del file di configurazione 6
Formato del file delle credenziali 10
Posizione dei file condivisi 10
Risoluzione della home directory 11
Cambia la posizione predefinita di questi file 12
Variabili di ambiente 13
Come impostare le variabili di ambiente 13
Configurazione delle variabili di ambiente senza server 14
Proprietà del sistema JVM 15
Come impostare le proprietà del sistema JVM 15
Autenticazione e accesso 18
ID Builder AWS 19
Autenticazione IAM Identity Center 20
Configura l'accesso programmatico utilizzando IAM Identity Center 20
Comprendi l'autenticazione IAM Identity Center 24
IAM Roles Anywhere 28
Fase 1: configurare IAM Roles Anywhere 28
Passaggio 2: utilizza IAM Roles Anywhere 28
Assunzione di un ruolo 30
Assumi un ruolo IAM 30
Federazione con identità web o OpenID Connect 32
AWS chiavi di accesso 33
Usa credenziali a breve termine 33
Usa credenziali a lungo termine 33
Credenziali a breve termine 35
Credenziali a lungo termine 36
Ruoli IAM per istanze Amazon EC2 39
Creazione di un ruolo IAM 39

Avvia un'istanza Amazon EC2 e specifica il tuo ruolo IAM 40
Connect all'istanza EC2 40
Esegui l'applicazione di esempio sull'istanza EC2 41
Riferimento alle impostazioni 42
Creazione di client di servizio 42
Precedenza delle impostazioni 42
`Config`elenco delle impostazioni dei file 44
`Credentials`elenco delle impostazioni dei file 47
elenco delle variabili di ambiente 48
Elenco delle proprietà del sistema JVM 52
Fornitori di credenziali standardizzati 54
Catena di fornitori di credenziali 55
AWS chiavi di accesso 56
Assumi il ruolo di fornitore 59
Fornitore di contenitori 66
Fornitore di IAM Identity Center 69
Fornitore IMDS 75
Fornitore di processi 80
Funzionalità standardizzate 84
ID dell'applicazione 85
Metadati delle istanze Amazon EC2. 87
Punti di accesso Amazon S3 89
Punti di accesso multi-Regione di Amazon S3 91
Regione AWS 93
AWS STS Endpoint regionalizzati 96
Endpoint dual-stack e FIPS 99
Rilevamento di endpoint 101
Configurazione generale 103
Cliente IMDS 106
Comportamento di ripetizione 109
Richiedi la compressione 115
Endpoint specifici del servizio 117
Impostazioni predefinite di configurazione intelligenti 166
Common Runtime 171
Dipendenze CRT 172
Politica di manutenzione 173

Panoramica 173
Controllo delle versioni 173
Ciclo di vita della versione principale dell'SDK 173
Ciclo di vita delle dipendenze 174
Metodi di comunicazione 175
Matrice di supporto delle versioni 176
Cronologia dei documenti 179
Glossario AWS 182
..... clxxxiii

AWS Guida di riferimento agli SDK e agli strumenti

Molti SDK e strumenti condividono alcune funzionalità comuni, tramite specifiche di progettazione condivise o tramite una libreria condivisa.

Questa guida include informazioni relative a:

- Configurazione— Come utilizzare le variabili condivise `config` e di `credentials` file o di ambiente per configurare gli AWS SDK e gli strumenti.
- Autenticazione e accesso— Stabilisci in che modo il codice o lo strumento si autentica AWS durante lo sviluppo con. Servizi AWS
- Riferimento alle impostazioni— Riferimento per tutte le impostazioni standardizzate disponibili per l'autenticazione e la configurazione.
- AWSLibrerie Common Runtime (CRT)— Panoramica delle librerie AWS Common Runtime (CRT) condivise disponibili per quasi tutti gli SDK.
- AWS Politica di manutenzione degli SDK e degli strumenticopre la politica di manutenzione e il controllo delle versioni per i AWS Software Development Kit (SDK) e gli strumenti, inclusi gli SDK per dispositivi mobili e Internet of Things (IoT), e le relative dipendenze sottostanti.

Questa guida di riferimento agli AWS SDK e agli strumenti vuole essere una base di informazioni applicabile a più SDK e strumenti. La guida specifica per l'SDK o lo strumento che stai utilizzando deve essere utilizzata in aggiunta a tutte le informazioni qui presentate. Di seguito sono riportati l'SDK e gli strumenti che contengono sezioni di materiale pertinenti in questa guida:

Se stai usando:	Le sezioni pertinenti di questa guida per te sono:
• Qualsiasi SDK o strumento	AWS Politica di manutenzione degli SDK e degli strumenti
• AWS Cloud Development Kit (AWS CDK) Guida per gli sviluppatori • AWS Serverless Application Model Guida per gli sviluppatori • AWS Toolkit for Eclipse Guida per l'utente	Configurazione Autenticazione e accesso AWS Politica di manutenzione degli SDK e degli strumenti

Se stai usando:	Le sezioni pertinenti di questa guida per te sono:
• AWS Toolkit for JetBrains Guida per l'utente • AWS Toolkit for Visual Studio Guida per l'utente • AWS Toolkit for Visual Studio Code Guida per l'utente	
• AWS Command Line Interface Guida per l'utente • AWS SDK for C++ Guida per gli sviluppatori • AWS SDK for Go Guida per gli sviluppatori • AWS SDK for Java Guida per gli sviluppatori • AWS SDK for JavaScript Guida per gli sviluppatori • SDK AWS for Kotlin • AWS SDK for .NET Guida per gli sviluppatori • AWS SDK for PHP Guida per gli sviluppatori • AWS SDK per Python (Boto3) - Guida introduttiva • AWS SDK for Ruby Guida per gli sviluppatori • AWS SDK for Rust • SDK AWS per Swift • AWS Tools for Windows PowerShell Guida per l'utente	Configurazione Autenticazione e accesso Riferimento alle impostazioni AWSLibrerie Common Runtime (CRT) AWS Politica di manutenzione degli SDK e degli strumenti AWS Matrice di supporto delle versioni degli SDK e degli strumenti

Risorse per sviluppatori

Per una panoramica degli strumenti che possono aiutarti a sviluppare applicazioni AWS, consulta Tools to Build on AWS. Per informazioni sul supporto, consulta il AWS Knowledge Center.

Amazon Q Developer è un assistente conversazionale generativo basato sull'intelligenza artificiale che può aiutarti a comprendere, creare, estendere e utilizzare le applicazioni. AWS Per accelerare

la tua crescita AWS, il modello alla base di Amazon Q è arricchito con AWS contenuti di alta qualità per produrre risposte più complete, utilizzabili e referenziate. Per ulteriori informazioni, consulta Cos'è Amazon Q Developer? nella Amazon Q Developer User Guide.

Notifica di telemetria del Toolkit

AWS I toolkit IDE (Integrated Development Environment) sono plugin ed estensioni che consentono l'accesso ai servizi dell'IDE. AWS Per informazioni dettagliate su ciascuno dei Toolkit IDE, consultate le Guide per l'utente del Toolkit nella tabella precedente.

AWS IDE Toolkit può raccogliere e archiviare dati di telemetria lato client per informare le decisioni relative alle future versioni del Toolkit. AWS I dati raccolti quantificano l'utilizzo del Toolkit. AWS

Per ulteriori informazioni sui dati di telemetria raccolti in tutti i toolkit AWS IDE, consulta il documento CommonDefinitions.json nel repository Github. `aws-toolkit-common`

Per informazioni dettagliate sui dati di telemetria raccolti da ciascuno dei Toolkit IDE, fai riferimento ai documenti di risorse nei seguenti repository Github di Toolkit: AWS AWS

- AWS Toolkit for Visual Studio
- AWS Toolkit for Visual Studio Code
- AWS Toolkit for JetBrains

Alcuni AWS servizi accessibili nei Toolkit possono raccogliere dati di telemetria aggiuntivi sul lato client. AWS Per informazioni dettagliate sul tipo di dati raccolti da ogni singolo AWS servizio, consulta l'argomento AWS Documentazione relativo al servizio specifico a cui sei interessato.

Configurazione

Con gli AWS SDK e altri strumenti di AWS sviluppo, come AWS Command Line Interface (AWS CLI), puoi interagire con le API di AWS servizio. Prima di eseguire questa operazione, tuttavia, è necessario configurare l'SDK o lo strumento con le informazioni necessarie per eseguire l'operazione richiesta.

Queste informazioni includono i seguenti elementi:

- Informazioni sulle credenziali che identificano chi sta chiamando l'API. Le credenziali vengono utilizzate per crittografare la richiesta ai server. AWS Utilizzando queste informazioni, AWS conferma la tua identità e puoi recuperare le politiche di autorizzazione ad essa associate. Quindi può determinare quali azioni sei autorizzato a eseguire.
- Altri dettagli di configurazione che usi per indicare all' AWS CLI SDK come elaborare la richiesta, dove inviare la richiesta (a quale endpoint del AWS servizio) e come interpretare o visualizzare la risposta.

Ogni SDK o strumento supporta più fonti che puoi utilizzare per fornire le credenziali e le informazioni di configurazione richieste. Alcune fonti sono esclusive dell'SDK o dello strumento e devi fare riferimento alla documentazione di tale strumento o SDK per i dettagli su come utilizzare tale metodo.

Tuttavia, la maggior parte degli AWS SDK e degli strumenti supporta impostazioni comuni provenienti da due fonti principali (oltre al codice stesso):

- File di AWS configurazione e credenziali condivisi: i **`credentials`** file `config` e condivisi sono il modo più comune per specificare l'autenticazione e la configurazione su un AWS SDK o uno strumento. Usa questi file per archiviare le impostazioni utilizzabili dai tuoi strumenti e dalle tue applicazioni. Le impostazioni all'interno `credentials` dei file condivisi `config` sono associate a un profilo specifico. Con più profili, puoi creare diverse configurazioni di impostazioni da applicare in diversi scenari. Quando utilizzate AWS uno strumento per richiamare un comando o utilizzate un SDK per richiamare un' AWS API, potete specificare quale profilo, e quindi quali impostazioni di configurazione, utilizzare per quell'azione. Uno dei profili è designato come `default` profilo e viene utilizzato automaticamente quando non si specifica esplicitamente un profilo da utilizzare. Le impostazioni che è possibile memorizzare in questi file sono documentate in questa guida di riferimento.
- Variabili di ambiente: in alternativa, alcune impostazioni possono essere memorizzate nelle variabili di ambiente del sistema operativo. Sebbene sia possibile avere un solo set di variabili di ambiente

attive alla volta, queste possono essere facilmente modificate dinamicamente man mano che il programma viene eseguito e i requisiti cambiano.

Argomenti aggiuntivi in questa sezione

- Condivisi config e credentials file
- Ubicazione degli elementi condivisi config e credentials dei file
- Supporto per variabili di ambiente
- Supporto delle proprietà del sistema JVM

Condivisi **`config`** e **`credentials`** file

I `credentials` file condivisi AWS `config` e contengono un set di profili. Un profilo è un insieme di impostazioni di configurazione, in coppie chiave-valore, utilizzato da AWS Command Line Interface (AWS CLI), dagli AWS SDK e da altri strumenti. I valori di configurazione sono allegati a un profilo per configurare alcuni aspetti dell'SDK/strumento quando viene utilizzato quel profilo. Questi file sono «condivisi» in quanto i valori hanno effetto su qualsiasi applicazione, processo o SDK nell'ambiente locale di un utente.

Sia i file condivisi che `config` i `credentials` file sono file di testo semplice che contengono solo caratteri ASCII (con codifica UTF-8). Assumono la forma di quelli che vengono generalmente definiti file INI.

Profili

Le impostazioni all'interno dei `credentials` file `config` e dei file condivisi sono associate a un profilo specifico. È possibile definire più profili all'interno del file per creare diverse configurazioni di impostazione da applicare in diversi ambienti di sviluppo.

Il `[default]` profilo contiene i valori utilizzati da un SDK o dall'operazione dello strumento se non viene specificato un profilo denominato specifico. Puoi anche creare profili separati a cui puoi fare riferimento esplicitamente per nome. Ogni profilo può utilizzare impostazioni e valori diversi in base alle esigenze dell'applicazione e dello scenario.

> **Note**
>
> `[default]`è semplicemente un profilo senza nome. Questo profilo è denominato `default` perché è il profilo predefinito utilizzato dall'SDK se l'utente non specifica un profilo. Non

fornisce valori predefiniti ereditati ad altri profili. Se si imposta qualcosa nel `[default]` profilo e non lo si imposta in un profilo denominato, il valore non viene impostato quando si utilizza il profilo denominato.

Imposta un profilo denominato

Il `[default]` profilo e più profili denominati possono esistere nello stesso file. Utilizza la seguente impostazione per selezionare le impostazioni del profilo utilizzate dall'SDK o dallo strumento durante l'esecuzione del codice. I profili possono anche essere selezionati all'interno del codice o per comando quando si lavora con. AWS CLI

Configura questa funzionalità impostando una delle seguenti opzioni:

`AWS_PROFILE`- variabile di ambiente

Quando questa variabile di ambiente è impostata su un profilo denominato o «predefinito», tutto il codice e AWS CLI i comandi SDK utilizzano le impostazioni di quel profilo.

Esempio in Linux/macOS di impostazione delle variabili di ambiente tramite riga di comando:

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

Esempio in Windows di impostazione delle variabili di ambiente tramite riga di comando:

```
setx AWS_PROFILE "my_default_profile_name"
```

`aws.profile`- Proprietà del sistema JVM

Per SDK for Kotlin su JVM e SDK for Java 2.x, puoi impostare la proprietà di sistema. `aws.profile` Quando l'SDK crea un client di servizio, utilizza le impostazioni nel profilo denominato a meno che l'impostazione non venga sovrascritta nel codice. L'SDK for Java 1.x non supporta questa proprietà di sistema.

Formato del file di configurazione

Il `config` file è organizzato in sezioni. Una sezione è una raccolta denominata di impostazioni e continua finché non viene incontrata un'altra riga di definizione della sezione.

Il `config` file è un file di testo semplice che utilizza il seguente formato:

- Tutte le voci di una sezione assumono la forma generale di. `setting-name=value`
- Le righe possono essere commentate iniziando la riga con un carattere hashtag ()#.

Tipi di sezione

Una definizione di sezione è una riga che applica un nome a una raccolta di impostazioni. Le linee di definizione della sezione iniziano e finiscono con parentesi quadre (`[]`). All'interno delle parentesi, c'è un identificatore del tipo di sezione e un nome personalizzato per la sezione. È possibile utilizzare lettere, numeri, trattini (`-`) e caratteri di sottolineatura (_), ma non spazi.

Tipo di sezione: **`default`**

Esempio di riga di definizione della sezione: `[default]`

`[default]`è l'unico profilo che non richiede l'identificatore di `profile` sezione.

L'esempio seguente mostra un `config` file di base con un `[default]` profilo. Imposta l'`region`impostazione. Tutte le impostazioni che seguono questa riga, fino alla definizione di un'altra sezione, fanno parte di questo profilo.

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

Tipo di sezione: **`profile`**

Esempio di riga di definizione della sezione: `[profile dev]`

La riga di definizione della `profile` sezione è un raggruppamento di configurazione denominato che è possibile applicare per diversi scenari di sviluppo. Per comprendere meglio i profili denominati, consultate la sezione precedente sui profili.

L'esempio seguente mostra un `config` file con una riga di definizione della `profile` sezione e un profilo denominato`foo`. Tutte le impostazioni che seguono questa riga, fino a quando non viene trovata un'altra definizione di sezione, fanno parte di questo profilo denominato.

```
[profile foo]
```

```
...settings...
```

Alcune impostazioni hanno un proprio gruppo annidato di sottoimpostazioni, come l's3impostazione e le sottoimpostazioni dell'esempio seguente. Associate le sottoimpostazioni al gruppo facendole rientrare con uno o più spazi.

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

Tipo di sezione: **`sso-session`**

Esempio di riga di definizione della sezione: `[sso-session my-sso]`

La riga di definizione della `sso-session` sezione nomina un gruppo di impostazioni utilizzate per configurare un profilo con cui risolvere AWS le credenziali. AWS IAM Identity Center Per ulteriori informazioni sulla configurazione dell'autenticazione Single Sign-On, vedere. Autenticazione IAM Identity Center Un profilo è collegato a una `sso-session` sezione tramite una coppia chiave-valore in cui `sso-session` è la chiave e il nome della `sso-session` sezione è il valore, ad esempio. `sso-session = <name-of-sso-session-section>`

L'esempio seguente configura un profilo che otterrà AWS le credenziali a breve termine per il ruolo "SampleRole" IAM nell'account «111122223333" utilizzando un token proveniente da «my-sso». La sezione «my-sso» viene referenziata `sso-session` nella sezione per nome utilizzando la chiave. `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
```

Tipo di sezione: **`services`**

Esempio di riga di definizione della sezione: `[services dev]`

> **Note**
>
> La `services` sezione supporta le personalizzazioni degli endpoint specifici del servizio ed è disponibile solo negli SDK e negli strumenti che includono questa funzionalità. Per vedere se questa funzionalità è disponibile per il tuo SDK, consulta la sezione dedicata agli endpoint specifici del servizio. Compatibilità con AWS gli SDK

La riga di definizione della `services` sezione indica un gruppo di impostazioni che configurano gli endpoint personalizzati per le richieste. Servizio AWS Un profilo è collegato a una `services` sezione da una coppia chiave-valore in cui `services` è la chiave e il nome della `services` sezione è il valore, ad esempio. `services = <name-of-services-section>`

La `services` sezione è ulteriormente suddivisa in sottosezioni mediante `<SERVICE> =` righe, dove si `<SERVICE>` trova la Servizio AWS chiave identificativa. L' Servizio AWS identificatore si basa sul modello API e sostituisce tutti gli spazi con caratteri `serviceId` di sottolineatura e tutte le lettere minuscole. Per un elenco di tutte le chiavi identificative del servizio da utilizzare nella sezione, vedere. `services` Identificatori per endpoint specifici del servizio La chiave identificativa del servizio è seguita da impostazioni annidate, ciascuna sulla propria riga e rientrata da due spazi.

L'esempio seguente utilizza una `services` definizione per configurare l'endpoint da utilizzare per le richieste effettuate solo al servizio. Amazon DynamoDB La `"local-dynamodb"` `services` sezione viene referenziata nella `profile` sezione per nome utilizzando la `services` chiave. La chiave Servizio AWS identificativa è. `dynamodb` La sottosezione del Amazon DynamoDB servizio inizia sulla linea. `dynamodb =` Tutte le righe immediatamente successive che sono rientrate sono incluse in tale sottosezione e si applicano a quel servizio.

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

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

Per ulteriori informazioni sulla configurazione personalizzata degli endpoint, vedere. Endpoint specifici del servizio

Formato del file delle credenziali

Le regole per il `credentials` file sono generalmente identiche a quelle del `config` file, tranne per il fatto che le sezioni del profilo non iniziano con la parola`profile`. Usa solo il nome del profilo stesso tra parentesi quadre. L'esempio seguente mostra un `credentials` file con una sezione di profilo denominata denominata. `foo`

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

Nel `credentials` file possono essere memorizzate solo le seguenti impostazioni considerate «segrete» o sensibili:`aws_access_key_id`,`aws_secret_access_key`, e`aws_session_token`. Sebbene queste impostazioni possano essere inserite in alternativa nel `config` file condiviso, ti consigliamo di conservare questi valori sensibili in un `credentials` file separato. In questo modo, puoi fornire autorizzazioni separate per ogni file, se necessario.

L'esempio seguente mostra un `credentials` file di base con un `[default]` profilo. Imposta `aws_access_key_id`e `aws_secret_access_key` le impostazioni e `aws_session_token` globali.

```
[default]
aws_access_key_id=AKIAIOSFODNN7EXAMPLE
aws_secret_access_key=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
aws_session_token=IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZVERYLONGSTRI
```

Indipendentemente dal fatto che si utilizzi un profilo denominato o `default` "" nel `credentials` file, tutte le impostazioni qui riportate verranno combinate con tutte le impostazioni `config` del file che utilizzano lo stesso nome di profilo. Se in entrambi i file sono presenti credenziali per un profilo che condivide lo stesso nome, le chiavi nel file delle credenziali hanno la precedenza.

Ubicazione degli elementi condivisi **`config`** e **`credentials`** dei file

I file condivisi AWS `config` e sono `credentials` file di testo semplice che contengono informazioni di configurazione per gli AWS SDK e gli strumenti. I file risiedono localmente nell'ambiente e vengono utilizzati automaticamente dal codice SDK o dai AWS CLI comandi eseguiti in tale ambiente. Ad esempio, sul tuo computer o durante lo sviluppo su un'istanza Amazon Elastic Compute Cloud.

Quando l'SDK o lo strumento vengono eseguiti, verifica la presenza di questi file e carica tutte le impostazioni di configurazione disponibili. Se i file non esistono già, un file di base viene creato automaticamente dall'SDK o dallo strumento.

Per impostazione predefinita, i file si trovano in una cartella denominata `.aws` che si trova nella cartella dell'utente home o dell'utente.

Sistema operativo	Posizione e nome predefiniti dei file
Linux e macOS	`~/.aws/config` `~/.aws/credentials`
Windows	`%USERPROFILE%\.aws\config` `%USERPROFILE%\.aws\credentials`

Risoluzione della home directory

~viene utilizzata per la risoluzione della home directory solo quando:

- Inizia il percorso
- È seguito immediatamente da `/` o da un separatore specifico della piattaforma. Su Windows, `~/` ed `~\` entrambi si risolvono nella home directory.

Quando si determina la home directory, vengono controllate le seguenti variabili:

- (Tutte le piattaforme) La variabile di `HOME` ambiente
- (Piattaforme Windows) La variabile di `USERPROFILE` ambiente
- (Piattaforme Windows) La concatenazione `HOMEDRIVE` e le variabili di `HOMEPATH` ambiente () `$HOMEDRIVE$HOMEPATH`
- (Opzionale per SDK o strumento) Una funzione o variabile di risoluzione del percorso home specifica dell'SDK o dello strumento

Quando possibile, se la home directory di un utente viene specificata all'inizio del percorso (ad esempio,`~username/`), viene risolta nella home directory del nome utente richiesto (ad esempio,). `/home/username/.aws/config`

Cambia la posizione predefinita di questi file

Puoi utilizzare una delle seguenti opzioni per sovrascrivere la posizione da cui questi file vengono caricati dall'SDK o dallo strumento.

Usa le variabili di ambiente

Le seguenti variabili di ambiente possono essere impostate per modificare la posizione o il nome di questi file dal valore predefinito a un valore personalizzato:

- `config`variabile di ambiente del file: **`AWS_CONFIG_FILE`**
- `credentials`variabile di ambiente di file: **`AWS_SHARED_CREDENTIALS_FILE`**

Linux/macOS

È possibile specificare una posizione alternativa eseguendo i seguenti comandi di esportazione su Linux o macOS.

```
$ 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

È possibile specificare una posizione alternativa eseguendo i seguenti comandi setx su Windows.

```
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
```

Per ulteriori informazioni sulla configurazione del sistema utilizzando le variabili di ambiente, vedere. Supporto per variabili di ambiente

Utilizzare le proprietà del sistema JVM

Per l'SDK per Kotlin in esecuzione su JVM e per l'SDK for Java 2.x, puoi impostare le seguenti proprietà del sistema JVM per modificare la posizione o il nome di questi file dal valore predefinito a un valore personalizzato:

- `config`proprietà del sistema JVM del file: **`aws.configFile`**

- `credentials`variabile di ambiente del file: **`aws.sharedCredentialsFile`**

Per istruzioni su come impostare le proprietà del sistema JVM, vedere. the section called “Come impostare le proprietà del sistema JVM” L'SDK for Java 1.x non supporta queste proprietà di sistema.

Supporto per variabili di ambiente

Le variabili di ambiente offrono un altro modo per specificare le opzioni di configurazione e le credenziali e possono essere utili per la creazione di script o l'impostazione temporanea di un profilo denominato come profilo di default. Per l'elenco delle variabili di ambiente supportate dalla maggior parte degli SDK, consulta. elenco delle variabili di ambiente

Precedenza delle opzioni

- Se specifichi un'impostazione utilizzando la relativa variabile di ambiente, questa sovrascrive qualsiasi valore caricato da un profilo nei file condivisi AWS `config` e. `credentials`
- Se specificate un'impostazione utilizzando un parametro sulla riga di AWS CLI comando, questo sovrascrive qualsiasi valore della variabile di ambiente corrispondente o di un profilo nel file di configurazione.

Come impostare le variabili di ambiente

L'esempio seguente mostra come configurare le variabili di ambiente per l'utente predefinito.

Linux, macOS, or Unix

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

L'impostazione della variabile di ambiente modifica il valore utilizzato fino al termine della sessione della shell o finché non imposti la variabile su un valore diverso. Puoi rendere le variabili persistenti per le sessioni future impostandole nello script di avvio della shell.

Windows Command Prompt

```
C:\> setx AWS_ACCESS_KEY_ID AKIAIOSFODNN7EXAMPLE
```

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

L'utilizzo `set` per impostare una variabile di ambiente modifica il valore utilizzato fino alla fine della sessione corrente del prompt dei comandi o fino a quando non si imposta la variabile su un valore diverso. Se si utilizza `setx`per impostare una variabile di ambiente, viene modificato il valore utilizzato sia nella sessione corrente del prompt dei comandi che in tutte le sessioni del prompt dei comandi create dopo l'esecuzione del comando. Ciò non ha alcun impatto su altre shell di comando già in esecuzione quando esegui il comando.

PowerShell

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

Se impostate una variabile di ambiente al PowerShell prompt, come mostrato negli esempi precedenti, il valore viene salvato solo per la durata della sessione corrente. Per rendere persistente l'impostazione della variabile di ambiente in tutte PowerShell le sessioni del prompt dei comandi, memorizzatela utilizzando l'applicazione System nel Pannello di controllo. In alternativa, puoi impostare la variabile per tutte le PowerShell sessioni future aggiungendola al tuo PowerShell profilo. Consulta la PowerShell documentazione per ulteriori informazioni sulla memorizzazione delle variabili di ambiente o sulla loro persistenza tra le sessioni.

Configurazione delle variabili di ambiente senza server

Se si utilizza un'architettura serverless per lo sviluppo, sono disponibili altre opzioni per l'impostazione delle variabili di ambiente. A seconda del contenitore, puoi utilizzare diverse strategie per l'esecuzione del codice in tali contenitori per visualizzare e accedere alle variabili di ambiente, in modo simile agli ambienti non cloud.

Ad esempio, conAWS Lambda, puoi impostare direttamente le variabili di ambiente. Per i dettagli, consulta Uso delle variabili di AWS Lambda ambiente nella Guida per AWS Lambda gli sviluppatori.

In Serverless Framework, puoi spesso impostare le variabili di ambiente SDK nel `serverless.yml` file sotto la chiave del provider sotto l'impostazione dell'ambiente. Per informazioni sul

`serverless.yml` file, consulta Impostazioni generali delle funzioni nella documentazione di Serverless Framework.

Indipendentemente dal meccanismo utilizzato per impostare le variabili di ambiente del contenitore, ce ne sono alcune riservate dal contenitore, come quelle documentate per Lambda nelle variabili di ambiente di runtime definite. Consulta sempre la documentazione ufficiale del contenitore che stai utilizzando per determinare come vengono trattate le variabili di ambiente e se esistono restrizioni.

Supporto delle proprietà del sistema JVM

Le proprietà del sistema JVM forniscono un altro modo per specificare le opzioni di configurazione e le credenziali per gli SDK che vengono eseguiti sulla JVM, ad esempio il e il. AWS SDK for Java SDK AWS for KotlinPer un elenco delle proprietà del sistema JVM supportate dagli SDK, consulta il riferimento alle impostazioni.

Precedenza delle opzioni

- Se specifichi un'impostazione utilizzando la relativa proprietà di sistema JVM, questa sovrascrive qualsiasi valore trovato nelle variabili di ambiente o caricato da un profilo in AWS e file condivisi. `config` `credentials`
- Se specifichi un'impostazione utilizzando la relativa variabile di ambiente, questa sovrascrive qualsiasi valore caricato da un profilo nei file `config` e `credentials` nei file AWS condivisi.

Come impostare le proprietà del sistema JVM

È possibile impostare le proprietà del sistema JVM in diversi modi.

Sulla riga di comando

Imposta le proprietà del sistema JVM sulla riga di comando quando richiami il `java` comando utilizzando lo switch. `-D` Il comando seguente configura Regione AWS globalmente per tutti i client di servizio a meno che non si sovrascriva esplicitamente il valore nel codice.

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

Se è necessario impostare più proprietà del sistema JVM, specificare lo switch più volte. `-D`

Con una variabile di ambiente

Se non riesci ad accedere alla riga di comando per richiamare la JVM per eseguire l'applicazione, puoi utilizzare la variabile di `JAVA_TOOL_OPTIONS` ambiente per configurare le opzioni della riga di comando. Questo approccio è utile in situazioni come l'esecuzione di una AWS Lambda funzione sul runtime Java o l'esecuzione di codice in una JVM incorporata.

L'esempio seguente configura Regione AWS globalmente per tutti i client di servizio a meno che non si sovrascriva esplicitamente il valore nel codice.

Linux, macOS, or Unix

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

L'impostazione della variabile di ambiente modifica il valore utilizzato fino al termine della sessione della shell o finché non imposti la variabile su un valore diverso. Puoi rendere le variabili persistenti per le sessioni future impostandole nello script di avvio della shell.

Windows Command Prompt

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

L'utilizzo `set` per impostare una variabile di ambiente modifica il valore utilizzato fino alla fine della sessione corrente del prompt dei comandi o fino a quando non si imposta la variabile su un valore diverso. Se si utilizza `setx`per impostare una variabile di ambiente, viene modificato il valore utilizzato sia nella sessione corrente del prompt dei comandi che in tutte le sessioni del prompt dei comandi create dopo l'esecuzione del comando. Ciò non ha alcun impatto su altre shell di comando già in esecuzione quando esegui il comando.

In fase di esecuzione

È inoltre possibile impostare le proprietà del sistema JVM in fase di esecuzione nel codice utilizzando il `System.setProperty` metodo illustrato nell'esempio seguente.

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

> **Important**
>
> Impostate le proprietà del sistema JVM prima di inizializzare i client del servizio SDK, altrimenti i client di servizio potrebbero utilizzare altri valori.

Autenticazione e accesso

È necessario stabilire in che modo il codice si autentica AWS durante lo sviluppo con. Servizi AWS È possibile configurare l'accesso programmatico alle AWS risorse in diversi modi, a seconda dell'ambiente e dell'AWSaccesso a disposizione.

Opzioni di autenticazione per il codice eseguito localmente (non inAWS)

- Autenticazione IAM Identity Center— Come best practice di sicurezza, ti consigliamo di utilizzare AWS Organizations IAM Identity Center per gestire l'accesso su tutti i tuoi dispositiviAccount AWS. Puoi creare utenti inAWS IAM Identity Center, utilizzare Microsoft Active Directory, utilizzare un provider di identità (IdP) SAML 2.0 o federare individualmente il tuo IdP in. Account AWS Per verificare se la tua regione supporta IAM Identity Center, consulta gli AWS IAM Identity Centerendpoint e le quote nel. Riferimenti generali di Amazon Web Services
- IAM Roles Anywhere— Puoi utilizzare IAM Roles Anywhere per ottenere credenziali di sicurezza temporanee in IAM per carichi di lavoro come server, contenitori e applicazioni eseguiti all'esterno di. AWS Per utilizzare IAM Roles Anywhere, i carichi di lavoro devono utilizzare certificati X.509.
- Assunzione di un ruolo— Puoi assumere un ruolo IAM per accedere temporaneamente a AWS risorse a cui altrimenti non avresti accesso.
- AWS chiavi di accesso— Altre opzioni che potrebbero essere meno convenienti o che potrebbero aumentare il rischio di sicurezza per le AWS risorse.

Opzioni di autenticazione per il codice in esecuzione all'interno di un AWS ambiente

- Utilizzo dei ruoli IAM per le istanze Amazon EC2— Usa i ruoli IAM per eseguire in modo sicuro la tua applicazione su un'istanza Amazon EC2.
- Puoi interagire a livello di codice con l'AWSutilizzo di IAM Identity Center nei seguenti modi:
  - AWS CloudShellDa utilizzare per eseguire AWS CLI comandi dalla console.
  - AWS Cloud9Da utilizzare per iniziare a programmare AWS utilizzando un ambiente di sviluppo integrato (IDE) con AWS risorse.
  - Per provare uno spazio di collaborazione basato sul cloud per i team di sviluppo software, prendi in considerazione l'utilizzo di Amazon. CodeCatalyst

Autenticazione tramite un provider di identità basato sul Web: applicazioni Web mobili o basate su client

Se stai creando applicazioni mobili o applicazioni web basate su client che richiedono l'accesso aAWS, crea la tua app in modo che richieda le credenziali di AWS sicurezza temporanee in modo dinamico utilizzando la federazione delle identità web.

Con la federazione delle identità Web, non è necessario creare il codice di accesso personalizzato o gestire le identità utente personalizzate. Gli utenti dell'app possono invece accedere utilizzando un provider di identità (IdP) esterno noto, come Login with Amazon, Facebook, Google o qualsiasi altro IdP compatibile con OpenID Connect (OIDC). Possono ricevere un token di autenticazione e scambiarsi quindi tale token per le credenziali di sicurezza temporanee in AWS che mappano a un ruolo IAM con autorizzazioni per utilizzare le risorse nel tuo Account AWS.

Per informazioni su come configurarlo per il tuo SDK o il tuo strumento, consulta. Federazione con identità web o OpenID Connect

Per le applicazioni mobili, prendi in considerazione l'utilizzo di Amazon Cognito. Amazon Cognito funge da broker di identità e svolge gran parte del lavoro federativo per te. Per ulteriori informazioni, consulta Using Amazon Cognito per app mobili nella IAM User Guide.

Ulteriori informazioni sulla gestione degli accessi

La Guida per l'utente IAM contiene le seguenti informazioni sul controllo sicuro dell'accesso alle AWS risorse:

- Identità IAM (utenti, gruppi di utenti e ruoli): scopri le basi delle identità in. AWS
- Best practice di sicurezza in IAM: raccomandazioni di sicurezza da seguire quando si sviluppano AWS applicazioni secondo il modello di responsabilità condivisa.

Riferimenti generali di Amazon Web ServicesHa le basi fondamentali su quanto segue:

- Comprensione e acquisizione AWS delle credenziali: accedi alle opzioni chiave e alle pratiche di gestione sia per l'accesso da console che per quello programmatico.

ID Builder AWS

I ID Builder AWS complementi Account AWS che possiedi già o che desideri creare. Sebbene un Account AWS funga da contenitore per AWS le risorse che crei e fornisca un limite di sicurezza per tali risorse, il tuo ti ID Builder AWS rappresenta come individuo. Puoi accedere con il tuo ID Builder AWS per accedere a strumenti e servizi per sviluppatori come Amazon CodeWhisperer e Amazon CodeCatalyst.

- Accedi tramite la ID Builder AWS Guida per l'Accedi ad AWSutente: scopri come creare e utilizzare un ID Builder ID Builder AWS e scopri cosa offre.
- Autenticazione con CodeWhisperer e Kit di strumenti AWS - Builder ID nella Guida per l'CodeWhisperer utente: scopri come utilizza un. CodeWhisperer ID Builder AWS
- CodeCatalystconcetti - ID Builder AWS nella Amazon CodeCatalyst User Guide - Scopri come CodeCatalyst usa unID Builder AWS.

Autenticazione IAM Identity Center

AWS IAM Identity Center è il metodo consigliato per fornire AWS credenziali quando si sviluppa su un servizio non di AWS elaborazione. Ad esempio, questo potrebbe essere qualcosa di simile al vostro ambiente di sviluppo locale. Se stai sviluppando su una AWS risorsa, come Amazon Elastic Compute Cloud (Amazon EC2) AWS Cloud9 oppure, ti consigliamo di ottenere le credenziali da quel servizio.

In questo tutorial, stabilisci l'accesso a IAM Identity Center e lo configurerai per il tuo SDK o strumento utilizzando il portale di AWS accesso e il. AWS CLI

- Il portale di AWS accesso è il luogo web in cui è possibile accedere manualmente a IAM Identity Center. Il formato dell'URL è `d-xxxxxxxxxx.awsapps.com/start` o*`your_subdomain`*`.awsapps.com/start`. Una volta effettuato l' AWS accesso al portale di accesso, è possibile visualizzare Account AWS i ruoli configurati per quell'utente. Questa procedura utilizza il portale di AWS accesso per ottenere i valori di configurazione necessari per il processo di autenticazione SDK/Tool.
- AWS CLI Viene utilizzato per configurare l'SDK o lo strumento per utilizzare l'autenticazione IAM Identity Center per le chiamate API effettuate dal codice. Questo processo monouso aggiorna il AWS `config` file condiviso, che viene poi utilizzato dal tuo SDK o dallo strumento quando esegui il codice.

Configura l'accesso programmatico utilizzando IAM Identity Center

Fase 1: Stabilire l'accesso e selezionare il set di autorizzazioni appropriato

Se non hai ancora abilitato IAM Identity Center, consulta Enabling IAM Identity Center nella Guida per l'AWS IAM Identity Center utente.

Scegli uno dei seguenti metodi per accedere alle tue AWS credenziali.

Non ho stabilito l'accesso tramite IAM Identity Center

1. Aggiungi un utente e aggiungi le autorizzazioni amministrative seguendo la procedura Configure user access with the default IAM Identity Center directory nella Guida per l'AWS IAM Identity Center utente.
2. Il set di `AdministratorAccess` autorizzazioni non deve essere utilizzato per lo sviluppo regolare. Si consiglia invece di utilizzare il set di `PowerUserAccess` autorizzazioni predefinito, a meno che il datore di lavoro non abbia creato un set di autorizzazioni personalizzato per questo scopo.

   Segui nuovamente la stessa procedura Configure user access with the IAM Identity Center directory predefinita, ma questa volta:

   - Invece di creare il *`Admin team`* gruppo, crea un *`Dev team`* gruppo e sostituiscilo successivamente nelle istruzioni.
   - È possibile utilizzare l'utente esistente, ma l'utente deve essere aggiunto al nuovo *`Dev team`* gruppo.
   - Invece di creare il set di *`AdministratorAccess`* autorizzazioni, create un set di *`PowerUserAccess`* autorizzazioni e sostituitelo successivamente nelle istruzioni.

   Quando hai finito, dovresti avere quanto segue:

   - Un `Dev team` gruppo.
   - Un set di `PowerUserAccess` autorizzazioni allegato al `Dev team` gruppo.
   - L'utente è stato aggiunto al `Dev team` gruppo.
3. Esci dal portale e accedi nuovamente per visualizzare le tue opzioni Account AWS e per `Administrator` or`PowerUserAccess`. Seleziona `PowerUserAccess` quando lavori con il tuo strumento/SDK.

Ho già accesso AWS tramite un provider di identità federato gestito dal mio datore di lavoro (come Microsoft Entra o Okta)

Accedi AWS tramite il portale del tuo provider di identità. Se il tuo amministratore cloud ti ha concesso le autorizzazioni `PowerUserAccess` (sviluppatore), vedi quelle a Account AWS cui hai accesso e il tuo set di autorizzazioni. Accanto al nome del set di autorizzazioni, vengono visualizzate

le opzioni per accedere agli account manualmente o programmaticamente utilizzando quel set di autorizzazioni.

Le implementazioni personalizzate potrebbero dare luogo a esperienze diverse, ad esempio nomi di set di autorizzazioni diversi. Se non sei sicuro del set di autorizzazioni da utilizzare, contatta il tuo team IT per ricevere assistenza.

Ho già accesso AWS tramite il portale di AWS accesso gestito dal mio datore di lavoro

Accedi AWS tramite il portale di AWS accesso. Se il tuo amministratore cloud ti ha concesso le autorizzazioni `PowerUserAccess` (sviluppatore), vedi quelle a Account AWS cui hai accesso e il tuo set di autorizzazioni. Accanto al nome del set di autorizzazioni, vengono visualizzate le opzioni per accedere agli account manualmente o programmaticamente utilizzando quel set di autorizzazioni.

Ho già accesso AWS tramite un provider di identità personalizzato federato gestito dal mio datore di lavoro

Contatta il tuo team IT per ricevere assistenza.

Passaggio 2: configura gli SDK e gli strumenti per utilizzare IAM Identity Center

1. Sul tuo computer di sviluppo, installa la versione più recente AWS CLI.

   a. Vedi Installazione o aggiornamento della versione più recente di AWS CLI nella Guida AWS Command Line Interface per l'utente.

   b. (Facoltativo) Per verificare che funzioni, aprite il prompt dei comandi ed eseguite il `aws --version` comando. AWS CLI

2. Accedere al portale di AWS accesso. Il tuo datore di lavoro può fornire questo URL o riceverlo tramite e-mail dopo la Fase 1: Stabilisci l'accesso. In caso contrario, trova l'URL del portale di AWS accesso nella dashboard di https://console.aws.amazon.com/singlesignon/.

   a. Nel portale di AWS accesso, nella scheda Account, seleziona il singolo account da gestire. Vengono visualizzati i ruoli dell'utente. Scegli le chiavi di accesso per ottenere le credenziali per la riga di comando o l'accesso programmatico per il set di autorizzazioni appropriato. Utilizza il set di `PowerUserAccess` autorizzazioni predefinito o qualsiasi set di autorizzazioni creato da te o dal tuo datore di lavoro per applicare le autorizzazioni con privilegi minimi per lo sviluppo.

   b. Nella finestra di dialogo Ottieni credenziali, scegli macOS e Linux o Windows, a seconda del sistema operativo in uso.

c. Scegli il metodo di credenziali IAM Identity Center per ottenere i `SSO Region` valori `SSO Start URL` e di cui hai bisogno per il passaggio successivo.

3. Nel prompt dei AWS CLI comandi, esegui il `aws configure sso` comando. Quando richiesto, inserisci i valori di configurazione raccolti nel passaggio precedente. Per i dettagli su questo AWS CLI comando, consulta Configurare il profilo con la `aws configure sso` procedura guidata.

   - Per il nome del profilo CLI, ti consigliamo di inserire il *`valore predefinito`* quando inizi. Per informazioni su come impostare profili non predefiniti (denominati) e la variabile di ambiente associata, consulta. Profili

4. (Facoltativo) Nel AWS CLI prompt dei comandi, confermate l'identità della sessione attiva eseguendo il `aws sts get-caller-identity` comando. La risposta dovrebbe mostrare il set di autorizzazioni IAM Identity Center che hai configurato.

5. Se utilizzi un AWS SDK, crea un'applicazione per il tuo SDK nel tuo ambiente di sviluppo.

   a. Per alcuni SDK, è `SSOOIDC` necessario aggiungere pacchetti aggiuntivi come `SSO` e all'applicazione prima di poter utilizzare l'autenticazione IAM Identity Center. Per i dettagli, consulta il tuo SDK specifico.

   b. Se in precedenza hai configurato l'accesso a AWS, esamina il AWS `credentials` file condiviso per verificarne l'eventuale AWS chiavi di accesso presenza. È necessario rimuovere tutte le credenziali statiche prima che l'SDK o lo strumento utilizzino le credenziali IAM Identity Center a causa della precedenza. Catena di fornitori di credenziali

Per un'analisi approfondita del modo in cui gli SDK e gli strumenti utilizzano e aggiornano le credenziali utilizzando questa configurazione, consulta. Comprendi l'autenticazione IAM Identity Center

A seconda della durata delle sessioni configurate, l'accesso alla fine scadrà e l'SDK o lo strumento riscontreranno un errore di autenticazione. Per aggiornare nuovamente la sessione del portale di accesso quando necessario, usa il comando AWS CLI per eseguire il comando. `aws sso login`

È possibile estendere sia la durata della sessione del portale di accesso IAM Identity Center sia la durata della sessione del set di autorizzazioni. In questo modo si allunga il periodo di tempo in cui è possibile eseguire il codice prima di dover accedere nuovamente manualmente con. AWS CLI Per ulteriori informazioni, consulta gli argomenti seguenti nella Guida per l'utente AWS IAM Identity Center :

- Durata della sessione di IAM Identity Center: configura la durata delle sessioni del portale di accesso degli utenti AWS
- Durata della sessione del set di autorizzazioni: imposta la durata della sessione

Per i dettagli su tutte le impostazioni del provider IAM Identity Center per SDK e strumenti, Provider di credenziali IAM Identity Center consulta questa guida.

Comprendi l'autenticazione IAM Identity Center

Termini pertinenti di IAM Identity Center

I seguenti termini aiutano a comprendere il processo e la configurazione alla baseAWS IAM Identity Center. La documentazione per le API AWS SDK utilizza nomi diversi da IAM Identity Center per alcuni di questi concetti di autenticazione. È utile conoscere entrambi i nomi.

La tabella seguente mostra come i nomi alternativi si relazionano tra loro.

Nome IAM Identity Center	Nome dell'API SDK	Description
Identity Center	`sso`	Sebbene AWS Single Sign-On venga rinominato, i namespace delle `sso` API manterranno il nome originale per motivi di compatibilità con le versioni precedenti. Per ulteriori informazioni, consulta la ridenominazione di IAM Identity Center nella Guida per l'utente. AWS IAM Identity Center
Console IAM Identity Center Console amministrativa		La console che usi per configurare il Single Sign-On.
AWSaccedere all'URL del portale		Un URL univoco per il tuo account IAM Identity Center, ad

Nome IAM Identity Center	Nome dell'API SDK	Description
		esempiohttps://*xxx*.awsapps.com/start . Accedi a questo portale utilizzando le tue credenziali di accesso IAM Identity Center.
Sessione IAM Identity Center Access Portal	Sessione di autenticazione	Fornisce un token di accesso al portatore al chiamante.
Sessione con set di autorizza zioni		La sessione IAM che l'SDK utilizza internamente per effettuare le Servizio AWS chiamate. Nelle discussio ni informali, potresti vederla erroneamente definita «sessione di ruolo».
Credenziali del set di autorizza zioni	AWSCredenziali credenziali sigv4	Le credenziali che l'SDK utilizza effettivamente per la maggior parte delle Servizio AWS chiamate (in particola re, tutte le chiamate sigv4). Servizio AWS Nelle discussio ni informali, potreste vederle erroneamente chiamate «credenziali di ruolo».
Provider di credenziali IAM Identity Center	Provider di credenziali SSO	Come si ottengono le credenziali, ad esempio la classe o il modulo che fornisce la funzionalità.

Comprendi la risoluzione delle credenziali SDK per Servizi AWS

L'API IAM Identity Center scambia le credenziali del token del portatore con credenziali sigv4. La maggior parte Servizi AWS sono API sigv4, con alcune eccezioni come e. Amazon CodeWhisperer

Amazon CodeCatalyst Di seguito viene descritto il processo di risoluzione delle credenziali utilizzato per supportare la maggior parte delle Servizio AWS chiamate per il codice dell'applicazione. AWS IAM Identity Center

Avvio di una sessione del portale di accesso AWS

- Inizia il processo accedendo alla sessione con le tue credenziali.
  - Usa il `aws sso login` comando in AWS Command Line Interface (AWS CLI). Questo avvia una nuova sessione di IAM Identity Center se non hai già una sessione attiva.
- Quando inizi una nuova sessione, ricevi un token di aggiornamento e un token di accesso da IAM Identity Center. AWS CLIInoltre aggiorna un file JSON della cache SSO con un nuovo token di accesso e un token di aggiornamento e lo rende disponibile per l'uso da parte degli SDK.
- Se hai già una sessione attiva, il AWS CLI comando riutilizza la sessione esistente e scadrà ogni volta che scade la sessione esistente. Per informazioni su come impostare la durata di una sessione di IAM Identity Center, consulta Configurare la durata delle sessioni del portale di AWS accesso degli utenti nella Guida per l'utente. AWS IAM Identity Center
  - La durata massima della sessione è stata estesa a 90 giorni per ridurre la necessità di accessi frequenti.

In che modo l'SDK ottiene le credenziali per le chiamate Servizio AWS

Gli SDK forniscono l'accesso a Servizi AWS quando si crea un'istanza di un oggetto client per servizio. Quando il profilo selezionato del AWS `config` file condiviso è configurato per la risoluzione delle credenziali di IAM Identity Center, IAM Identity Center viene utilizzato per risolvere le credenziali per l'applicazione.

- Il processo di risoluzione delle credenziali viene completato durante l'esecuzione, quando viene creato un client.

Per recuperare le credenziali per le API sigv4 utilizzando il single sign-on di IAM Identity Center, l'SDK utilizza il token di accesso IAM Identity Center per ottenere una sessione IAM. Questa sessione IAM è chiamata sessione del set di autorizzazioni e fornisce l'AWSaccesso all'SDK assumendo un ruolo IAM.

- La durata della sessione del set di autorizzazioni è impostata indipendentemente dalla durata della sessione di IAM Identity Center.

  - Per informazioni su come impostare la durata della sessione del set di autorizzazioni, consulta Impostare la durata della sessione nella Guida per l'AWS IAM Identity Centerutente.
- Tieni presente che le credenziali del set di autorizzazioni vengono anche chiamate credenziali e AWScredenziali sigv4 nella maggior parte della documentazione sulle API SDK. AWS

Le credenziali del set di autorizzazioni vengono restituite da una chiamata all'API IAM Identity getRoleCredentialsCenter all'SDK. L'oggetto client dell'SDK utilizza quel ruolo IAM assunto per effettuare chiamate aServizio AWS, ad esempio chiedere ad Amazon S3 di elencare i bucket nel tuo account. L'oggetto client può continuare a funzionare utilizzando le credenziali del set di autorizzazioni fino alla scadenza della sessione del set di autorizzazioni.

Scadenza e aggiornamento della sessione

Quando si utilizza ilConfigurazione del provider di token SSO, il token di accesso orario ottenuto da IAM Identity Center viene aggiornato automaticamente utilizzando il token di aggiornamento.

- Se il token di accesso è scaduto quando l'SDK tenta di utilizzarlo, l'SDK utilizza il token di aggiornamento per cercare di ottenere un nuovo token di accesso. IAM Identity Center confronta il token di aggiornamento con la durata della sessione del portale di accesso IAM Identity Center. Se il token di aggiornamento non è scaduto, IAM Identity Center risponde con un altro token di accesso.
- Questo token di accesso può essere utilizzato per aggiornare la sessione del set di autorizzazioni dei client esistenti o per risolvere le credenziali per nuovi client.

Tuttavia, se la sessione del portale di accesso IAM Identity Center è scaduta, non viene concesso alcun nuovo token di accesso. Pertanto, la durata del set di autorizzazioni non può essere rinnovata. Scadrà (e l'accesso verrà perso) ogni volta che scade la durata della sessione del set di autorizzazioni memorizzato nella cache per i client esistenti.

Qualsiasi codice che crea un nuovo client fallirà l'autenticazione non appena scadrà la sessione di IAM Identity Center. Questo perché le credenziali del set di autorizzazioni non vengono memorizzate nella cache. Il codice non sarà in grado di creare un nuovo client e completare il processo di risoluzione delle credenziali finché non avrai un token di accesso valido.

Ricapitolando, quando l'SDK necessita di nuove credenziali del set di autorizzazioni, l'SDK verifica innanzitutto la presenza di eventuali credenziali valide ed esistenti e le utilizza. Questo vale sia che le credenziali siano per un nuovo client o per un client esistente con credenziali scadute. Se

le credenziali non vengono trovate o non sono valide, l'SDK chiama l'API IAM Identity Center per ottenere nuove credenziali. Per chiamare l'API, è necessario il token di accesso. Se il token di accesso è scaduto, l'SDK utilizza il token di aggiornamento per cercare di ottenere un nuovo token di accesso dal servizio IAM Identity Center. Questo token viene concesso se la sessione del portale di accesso IAM Identity Center non è scaduta.

IAM Roles Anywhere

Puoi utilizzare IAM Roles Anywhere per ottenere credenziali di sicurezza temporanee in IAM per carichi di lavoro come server, contenitori e applicazioni eseguiti all'esterno di. AWS Per utilizzare IAM Roles Anywhere, i carichi di lavoro devono utilizzare certificati X.509. L'amministratore cloud deve fornire il certificato e la chiave privata necessari per configurare IAM Roles Anywhere come fornitore di credenziali.

Fase 1: configurare IAM Roles Anywhere

IAM Roles Anywhere offre un modo per ottenere credenziali temporanee per un carico di lavoro o un processo eseguito all'esterno di. AWS Viene stabilito un trust anchor con l'autorità di certificazione per ottenere credenziali temporanee per il ruolo IAM associato. Il ruolo imposta le autorizzazioni che il carico di lavoro avrà quando il codice si autentica con IAM Roles Anywhere.

Per i passaggi per configurare trust anchor, IAM role e IAM Roles Anywhere, consulta Creating a trust anchor and profile in Roles Anywhere nella Guida per l'utente di IAM AWS Identity and Access Management Roles Anywhere.

> **Note**
>
> Un profilo nella IAM Roles Anywhere User Guide si riferisce a un concetto unico all'interno del servizio IAM Roles Anywhere. Non è correlato ai profili all'interno del AWS `config` file condiviso.

Passaggio 2: utilizza IAM Roles Anywhere

Per ottenere credenziali di sicurezza temporanee da IAM Roles Anywhere, utilizza lo strumento di supporto alle credenziali fornito da IAM Roles Anywhere. Lo strumento per le credenziali implementa il processo di firma per IAM Roles Anywhere.

Per istruzioni su come scaricare lo strumento di supporto alle credenziali, consulta Ottenere credenziali di sicurezza temporanee da AWS Identity and Access Management Roles Anywhere nella Guida per l'utente di IAM Roles Anywhere.

Per utilizzare le credenziali di sicurezza temporanee di IAM Roles Anywhere con AWS SDK eAWS CLI, puoi configurare le `credential_process` impostazioni nel file condiviso. AWS `config` Gli SDK e AWS CLI supportano un provider di credenziali di processo che utilizza per l'autenticazione. `credential_process` Di seguito viene illustrata la struttura generale da impostare. `credential_process`

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

Il `credential-process` comando dello strumento di supporto restituisce credenziali temporanee in un formato JSON standard compatibile con l'impostazione. `credential_process` Notate che il nome del comando contiene un trattino, ma il nome dell'impostazione contiene un carattere di sottolineatura. Il comando richiede i seguenti parametri:

- `private-key`— Il percorso della chiave privata che ha firmato la richiesta.
- `certificate`— Il percorso del certificato.
- `role-arn`— L'ARN del ruolo per cui ottenere le credenziali temporanee.
- `profile-arn`— L'ARN del profilo che fornisce una mappatura per il ruolo specificato.
- `trust-anchor-arn`— L'ARN del trust anchor utilizzato per l'autenticazione.

L'amministratore del cloud deve fornire il certificato e la chiave privata. Tutti e tre i valori ARN possono essere copiati da. AWS Management Console L'esempio seguente mostra un `config` file condiviso che configura il recupero delle credenziali temporanee dallo strumento di supporto.

```
[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
```

Per i parametri opzionali e i dettagli aggiuntivi dello strumento di supporto, consulta IAM Roles Anywhere Credential Helper on. GitHub

Per i dettagli sull'impostazione della configurazione SDK stessa e sul fornitore delle credenziali di processo, consulta questa guida. Provider di credenziali di processo

Assunzione di un ruolo

Assumere un ruolo implica l'utilizzo di un set di credenziali di sicurezza temporanee per accedere a AWS risorse a cui altrimenti non avreste accesso. Le credenziali temporanee sono costituite da un ID chiave di accesso, una chiave di accesso segreta e un token di sicurezza. Per ulteriori informazioni sulle richieste API AWS Security Token Service (AWS STS), consulta Azioni nell'AWS Security Token ServiceAPI Reference.

Per configurare l'SDK o lo strumento per assumere un ruolo, devi prima creare o identificare un ruolo specifico da assumere. I ruoli IAM sono identificati in modo univoco da un ruolo Amazon Resource Name (ARN). I ruoli stabiliscono relazioni di fiducia con un'altra entità. L'entità affidabile che utilizza il ruolo potrebbe essere un'altra Servizio AWSAccount AWS, un provider di identità Web o una federazione OIDC o SAML. Per ulteriori informazioni sui ruoli IAM, consulta Using IAM roles nella IAM User Guide.

Dopo aver identificato il ruolo IAM, se quel ruolo ti affida la fiducia, puoi configurare il tuo SDK o lo strumento per utilizzare le autorizzazioni concesse dal ruolo. Per farlo, o. Assumi un ruolo IAM Federazione con identità web o OpenID Connect

Assumi un ruolo IAM

Quando assume un ruolo, AWS STS restituisce un set di credenziali di sicurezza temporanee. Queste credenziali provengono da un altro profilo o dall'istanza o dal contenitore in cui è in esecuzione il codice. Altri esempi di assunzione di un ruolo includono la gestione di più account Account AWS da Amazon EC2, l'AWS CodeCommitutilizzo Account AWS di più account o l'accesso a un altro account da. AWS CodeBuild

Fase 1: configurare un ruolo IAM

Per configurare il tuo SDK o lo strumento per assumere un ruolo, devi prima creare o identificare un ruolo specifico da assumere. I ruoli IAM vengono identificati in modo univoco utilizzando un ruolo ARN. I ruoli stabiliscono relazioni di fiducia con un'altra entità, in genere all'interno dell'account o per l'accesso tra account. Per configurarlo, consulta Creazione di ruoli IAM nella Guida per l'utente IAM.

Passaggio 2: configura l'SDK o lo strumento

Configura l'SDK o lo strumento per ottenere le credenziali da o. `credential_source` `source_profile`

`credential_source`Utilizzalo per ottenere credenziali da un contenitore Amazon ECS, un'istanza Amazon EC2 o da variabili di ambiente.

Utilizza `source_profile` per ottenere credenziali da un altro profilo. `source_profile`supporta anche il concatenamento dei ruoli, ossia gerarchie di profili in cui un ruolo assunto viene poi utilizzato per assumere un altro ruolo.

Quando lo specificate in un profilo, l'SDK o lo strumento effettua automaticamente la chiamata AWS STS `AssumeRole`API corrispondente. Per recuperare e utilizzare credenziali temporanee assumendo un ruolo, specificate i seguenti valori di configurazione nel file condiviso. AWS `config` Per maggiori dettagli su ciascuna di queste impostazioni, consulta la sezione. Assumi le impostazioni del fornitore di credenziali di ruolo

- `role_arn`- Dal ruolo IAM che hai creato nella fase 1
- Configura uno `source_profile` o `credential_source`
- (Facoltativo) `duration_seconds`
- (Facoltativo) `external_id`
- (Facoltativo) `mfa_serial`
- (Facoltativo) `role_session_name`

Gli esempi seguenti mostrano la configurazione di entrambe le opzioni di assunzione del ruolo in un `config` file condiviso:

```
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
```

Per i dettagli su tutte le impostazioni del provider di credenziali di assunzione del ruolo, Assumi il ruolo di fornitore di credenziali consulta questa guida.

Federazione con identità web o OpenID Connect

Quando si creano applicazioni mobili o applicazioni Web basate su client che richiedono l'accesso aAWS, AWS STS restituisce un set di credenziali di sicurezza temporanee per gli utenti federati autenticati tramite un provider di identità pubblico (IdP). Esempi di provider di identità pubblici comprendono Login with Amazon, Facebook, Google o qualsiasi provider di identità compatibile con OpenID Connect (OIDC). Con questo metodo, gli utenti non hanno bisogno delle proprie identità o di quelle di IAM. AWS

Se utilizzi Amazon Elastic Kubernetes Service, questa funzionalità offre la possibilità di specificare diversi ruoli IAM per ciascuno dei tuoi contenitori. Kubernetes offre la possibilità di distribuire token OIDC ai contenitori, che vengono utilizzati da questo fornitore di credenziali per ottenere credenziali temporanee. Per ulteriori informazioni su questa configurazione di Amazon EKS, consulta i ruoli IAM per gli account di servizio nella Amazon EKS User Guide. Tuttavia, per un'opzione più semplice, ti consigliamo di utilizzare invece Amazon EKS Pod Identities se il tuo SDK lo supporta.

Fase 1: configurare un provider di identità e un ruolo IAM

Per configurare la federazione con un IdP esterno, utilizza un provider di identità IAM per fornire AWS informazioni sull'IdP esterno e sulla sua configurazione. In questo modo si instaura un rapporto di fiducia tra il tuo Account AWS e l'IdP esterno. Prima di configurare l'SDK per utilizzare il token di identità web per l'autenticazione, devi prima configurare il provider di identità (IdP) e il ruolo IAM utilizzato per accedervi. Per configurarli, consulta Creating a role for web identity o OpenID Connect Federation (console) nella IAM User Guide.

Passaggio 2: configura l'SDK o lo strumento

Configura l'SDK o lo strumento per utilizzare un token di identità Web AWS STS per l'autenticazione.

Quando lo specifichi in un profilo, l'SDK o lo strumento effettua automaticamente la chiamata AWS STS `AssumeRoleWithWebIdentity`API corrispondente per te. Per recuperare e utilizzare le credenziali temporanee utilizzando la federazione delle identità Web, specificate i seguenti valori di configurazione nel file condiviso. AWS `config` Per maggiori dettagli su ciascuna di queste impostazioni, consulta la Assumi le impostazioni del fornitore di credenziali di ruolo sezione.

- `role_arn`- Dal ruolo IAM che hai creato nella fase 1
- `web_identity_token_file`- Dall'IdP esterno
- (Facoltativo) `duration_seconds`
- (Facoltativo) `role_session_name`

Di seguito è riportato un esempio di configurazione di `config` file condivisa per assumere un ruolo con identità web:

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

> **Note**
>
> Per le applicazioni mobili, prendi in considerazione l'utilizzo di Amazon Cognito. Amazon Cognito funge da broker di identità e svolge gran parte del lavoro federativo per te. Tuttavia, il provider di identità Amazon Cognito non è incluso nelle librerie di base degli SDK e degli strumenti come altri provider di identità. Per accedere all'API Amazon Cognito, includi il client del servizio Amazon Cognito nella build o nelle librerie del tuo SDK o strumento. Per l'utilizzo con AWS gli SDK, consulta Esempi di codice nella Amazon Cognito Developer Guide.

Per i dettagli su tutte le impostazioni del provider di credenziali di assunzione del ruolo, consulta questa guidaAssumi il ruolo di fornitore di credenziali.

AWS chiavi di accesso

Usa credenziali a breve termine

Ti consigliamo di configurare l'SDK o lo strumento da utilizzare per utilizzare Autenticazione IAM Identity Center opzioni di durata estesa della sessione.

Tuttavia, per configurare direttamente l'SDK o le credenziali temporanee dello strumento, consulta. Autenticazione utilizzando credenziali a breve termine

Usa credenziali a lungo termine

> **Warning**
>
> Per evitare rischi per la sicurezza, non utilizzare gli utenti IAM per l'autenticazione quando sviluppi software creato ad hoc o lavori con dati reali. Utilizza invece la federazione con un provider di identità come AWS IAM Identity Center.

Gestisci l'accesso in tutto Account AWS

Come best practice di sicurezza, ti consigliamo di utilizzare AWS Organizations IAM Identity Center per gestire l'accesso su tutti i tuoi Account AWS. Per ulteriori informazioni, consulta Best Practice di sicurezza in IAM nella Guida per l'utente di IAM.

Puoi creare utenti in IAM Identity Center, utilizzare Microsoft Active Directory, utilizzare un provider di identità (IdP) SAML 2.0 o federare individualmente il tuo IdP. Account AWSUtilizzando uno di questi approcci, puoi fornire un'esperienza Single Sign-On ai tuoi utenti. Puoi anche applicare l'autenticazione a più fattori (MFA) e utilizzare credenziali temporanee per l'accesso. Account AWS Ciò differisce da un utente IAM, che è una credenziale a lungo termine che può essere condivisa e che potrebbe aumentare il rischio di sicurezza per le risorse. AWS

Crea utenti IAM solo per ambienti sandbox

Se sei un principiante AWS, potresti creare un utente IAM di prova e poi utilizzarlo per eseguire tutorial ed esplorare ciò che AWS ha da offrire. Va bene usare questo tipo di credenziale quando impari, ma ti consigliamo di evitare di utilizzarle al di fuori di un ambiente sandbox.

Per i seguenti casi d'uso, potrebbe essere utile iniziare con gli utenti IAM in: AWS

- Inizia a usare il tuo AWS SDK o il tuo strumento ed esplora Servizi AWS in un ambiente sandbox.
- L'esecuzione di script, processi e altri processi automatizzati pianificati che non supportano un processo di accesso con assistenza umana come parte del tuo apprendimento.

Se utilizzi utenti IAM al di fuori di questi casi d'uso, passa a IAM Identity Center o federa il tuo provider di identità a Account AWS il prima possibile. Per ulteriori informazioni, consulta Identity Federation in AWS.

Chiavi di accesso utente IAM sicure

È necessario ruotare regolarmente le chiavi di accesso utente IAM. Segui le indicazioni contenute nella sezione Rotating access keys nella IAM User Guide. Se ritieni di aver condiviso accidentalmente le tue chiavi di accesso utente IAM, ruota le chiavi di accesso.

Le chiavi di accesso utente IAM devono essere archiviate nel AWS `credentials` file condiviso sul computer locale. Non memorizzate le chiavi di accesso utente IAM nel codice. Non includere file di configurazione che contengono le chiavi di accesso utente IAM all'interno di alcun software di

gestione del codice sorgente. Strumenti esterni, come il progetto open source git-secrets, possono aiutarti a non inserire inavvertitamente informazioni sensibili in un repository Git. Per ulteriori informazioni, consulta IAM Identities (users, user groups, and roles) nella IAM User Guide.

Per configurare un utente IAM per iniziare, consulta Autenticazione utilizzando credenziali a lungo termine.

Autenticazione utilizzando credenziali a breve termine

Ti consigliamo di configurare l'SDK o lo strumento da utilizzare Autenticazione IAM Identity Center con opzioni di durata prolungata della sessione. Tuttavia, puoi copiare e utilizzare le credenziali temporanee disponibili nel portale di accesso. AWS Le nuove credenziali dovranno essere copiate quando scadono. È possibile utilizzare le credenziali temporanee in un profilo o utilizzarle come valori per le proprietà di sistema e le variabili di ambiente.

Configura un file di credenziali utilizzando credenziali a breve termine recuperate dal portale di accesso AWS

1. Crea un file di credenziali condiviso.
2. Nel file delle credenziali, incolla il seguente testo segnaposto fino a incollare le credenziali temporanee di lavoro.

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

3. Salvare il file. Il file `~/.aws/credentials` dovrebbe ora esistere sul tuo sistema di sviluppo locale. Questo file contiene il profilo [predefinito] utilizzato dall'SDK o dallo strumento se non viene specificato un profilo denominato specifico.
4. Accedi al portale di AWS accesso.
5. Segui queste istruzioni per l'aggiornamento manuale delle credenziali per copiare le credenziali del ruolo IAM dal AWS portale di accesso.

   a. Per la fase 4 delle istruzioni collegate, scegli il nome del ruolo IAM che concede l'accesso per le tue esigenze di sviluppo. Questo ruolo in genere ha un nome simile PowerUserAccessa Developer.

   b. Per il passaggio 7 delle istruzioni collegate, seleziona l'opzione Aggiungi manualmente un profilo al file delle AWS credenziali e copia il contenuto.

6. Incolla le credenziali copiate nel file locale. `credentials` Il nome del profilo generato non è necessario se si utilizza il `default` profilo. Il file dovrebbe essere simile al seguente.

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

7. Salvare il file `credentials`.

Quando l'SDK crea un client di servizio, accederà a queste credenziali temporanee e le utilizzerà per ogni richiesta. Le impostazioni per il ruolo IAM scelto nella fase 5a determinano la durata di validità delle credenziali temporanee. La durata massima è di dodici ore.

Dopo la scadenza delle credenziali temporanee, ripeti i passaggi da 4 a 7.

Autenticazione utilizzando credenziali a lungo termine

> **Warning**
>
> Per evitare rischi per la sicurezza, non utilizzare gli utenti IAM per l'autenticazione quando sviluppi software creato ad hoc o lavori con dati reali. Utilizza invece la federazione con un provider di identità come AWS IAM Identity Center.

Se utilizzi un utente IAM per eseguire il codice, l'SDK o lo strumento nel tuo ambiente di sviluppo si autentica utilizzando credenziali utente IAM a lungo termine nel file condiviso. AWS `credentials` Consulta le best practice di sicurezza nell'argomento IAM e passa a IAM Identity Center o ad altre credenziali temporanee il prima possibile.

Avvertenze e linee guida importanti per le credenziali

Avvertenze per le credenziali

- NON utilizzare le credenziali root dell'account per accedere alle risorse AWS . Queste credenziali forniscono un accesso illimitato all'account e sono difficili da revocare.
- NON inserire chiavi di accesso letterali o informazioni sulle credenziali nei file dell'applicazione. In caso contrario, rischi di esporre accidentalmente le credenziali se, per esempio, carichi il progetto in repository pubblici.

- NON includete file che contengono credenziali nell'area del progetto.
- Tieni presente che tutte le credenziali memorizzate nel AWS `credentials` file condiviso vengono archiviate in testo non crittografato.

Linee guida aggiuntive per la gestione sicura delle credenziali

Per una discussione generale su come gestire in modo sicuro le AWS credenziali, vedere Procedure ottimali per la gestione AWS delle chiavi di accesso nel. Riferimenti generali di AWS Considera inoltre quanto segue:

- Usa ruoli IAM per le attività di Amazon Elastic Container Service (Amazon ECS).
- Usa ruoli IAM per le applicazioni in esecuzione sulle istanze Amazon EC2.

Prerequisiti: creare un account AWS

Per utilizzare un utente IAM per accedere ai AWS servizi, sono necessari un AWS account e delle AWS credenziali.

1. Creazione di un account.

   Per creare un AWS account, vedi Guida introduttiva: sei un utente principiante AWS? nella Guida di AWS Account Management riferimento.

2. Creazione di un utente amministratore.

   Evita di utilizzare l'account utente root (l'account iniziale creato) per accedere alla console di gestione e ai servizi. Crea invece un account utente amministratore, come illustrato in Creazione di un utente amministratore nella Guida per l'utente di IAM.

   Dopo aver creato l'account utente amministratore e registrato i dettagli di accesso, assicurati di disconnetterti dall'account utente root e di accedere nuovamente utilizzando l'account amministrativo.

Nessuno di questi account è adatto per lo sviluppo AWS o l'esecuzione di applicazioni su AWS. Come procedura ottimale, è necessario creare utenti, set di autorizzazioni o ruoli di servizio appropriati per queste attività. Per ulteriori informazioni, consulta Applicazione delle autorizzazioni del privilegio minimo nella Guida per l'utente di IAM.

Passaggio 1: crea il tuo utente IAM

- Crea il tuo utente IAM seguendo la procedura Creating IAM users (console) nella IAM User Guide. Quando crei il tuo utente IAM:
  - Ti consigliamo di selezionare Fornisci l'accesso utente a AWS Management Console. Ciò ti consente di visualizzare Servizi AWS il codice in esecuzione in un ambiente visivo, ad esempio controllando i log di AWS CloudTrail diagnostica o caricando file su Amazon Simple Storage Service, il che è utile per il debug del codice.
  - Per le opzioni Imposta autorizzazioni - Autorizzazione, seleziona Allega direttamente le politiche per indicare come desideri assegnare le autorizzazioni a questo utente.
    - La maggior parte dei tutorial SDK «Getting Started» utilizza il servizio Amazon S3 come esempio. Per fornire alla tua applicazione l'accesso completo ad Amazon S3, seleziona la `AmazonS3FullAccess` policy da allegare a questo utente.
  - Puoi ignorare i passaggi facoltativi di tale procedura relativi all'impostazione dei limiti o dei tag di autorizzazione.

Passaggio 2: Ottieni le tue chiavi di accesso

1. Nel pannello di navigazione della console IAM, seleziona Utenti, quindi seleziona **`User name`** l'utente che hai creato in precedenza.
2. Nella pagina dell'utente, seleziona la pagina Credenziali di sicurezza. Quindi, in Chiavi di accesso, seleziona Crea chiave di accesso.
3. Per Creare la chiave di accesso Step 1, scegli Command Line Interface (CLI) o Codice locale. Entrambe le opzioni generano lo stesso tipo di chiave da utilizzare sia con gli SDK che con AWS CLI gli SDK.
4. Per la creazione della chiave di accesso (Fase 2), inserisci un tag opzionale e seleziona Avanti.
5. Per la fase 3 di creazione della chiave di accesso, seleziona Scarica il file.csv per salvare un **`.csv`** file con la chiave di accesso e la chiave di accesso segreta del tuo utente IAM. Ti serviranno queste informazioni per dopo.

   > **Warning**
   >
   > Utilizza le misure di sicurezza appropriate per proteggere queste credenziali.

6. Seleziona Done (Fatto)

Passaggio 3: Aggiornare il file condiviso **`credentials`**

1. Crea o apri il AWS `credentials` file condiviso. Questo file si trova `~/.aws/credentials` su sistemi Linux e macOS e `%USERPROFILE%\.aws\credentials` su Windows. Per ulteriori informazioni, consulta Posizione dei file delle credenziali.
2. Aggiungi il testo seguente al `credentials` file condiviso. Sostituisci il valore ID di esempio e il valore della chiave di esempio con i valori del `.csv` file scaricato in precedenza.

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

3. Salvare il file.

Il `credentials` file condiviso è il modo più comune per archiviare le credenziali. Queste possono anche essere impostate come variabili di ambiente, vedi AWS chiavi di accesso per i nomi delle variabili di ambiente. Questo è un modo per iniziare, ma ti consigliamo di passare a IAM Identity Center o ad altre credenziali temporanee il prima possibile. Dopo aver abbandonato l'utilizzo di credenziali a lungo termine, ricordati di eliminare queste credenziali dal file condiviso. `credentials`

Utilizzo dei ruoli IAM per le istanze Amazon EC2

Questo esempio illustra la configurazione di un AWS Identity and Access Management ruolo con accesso Amazon S3 da utilizzare nell'applicazione distribuita su un'istanza Amazon EC2.

Per un'istanza Amazon Elastic Compute Cloud, crea un ruolo IAM, quindi consenti alla tua istanza Amazon EC2 di accedere a quel ruolo. Per ulteriori informazioni, consulta IAM Roles for Amazon EC2 nella Amazon EC2 User Guide o IAM Roles for Amazon EC2 nella Amazon EC2 User Guide.

Creazione di un ruolo IAM

Crea un ruolo IAM che garantisca l'accesso in sola lettura ad Amazon S3.

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, seleziona Ruoli, quindi seleziona Crea ruolo.
3. Per Seleziona entità attendibile, in Tipo di entità affidabile, scegli Servizio AWS.

4. In Caso d'uso, scegli Amazon EC2, quindi seleziona Avanti.
5. Per Aggiungi autorizzazioni, seleziona la casella di controllo per Amazon S3 Read Only Access dall'elenco delle policy, quindi seleziona Avanti.
6. Inserisci un nome per il ruolo, quindi seleziona Crea ruolo. Ricorda questo nome perché ti servirà quando lancerai la tua istanza Amazon EC2.

Avvia un'istanza Amazon EC2 e specifica il tuo ruolo IAM

Puoi avviare un'istanza Amazon EC2 con un ruolo IAM utilizzando la console Amazon EC2.

Segui le istruzioni per avviare un'istanza nella Amazon EC2 User Guide o nella Amazon EC2 User Guide.

Quando raggiungi la pagina Review Instance Launch (Verifica del lancio dell'istanza), seleziona Edit instance details (Modifica dettagli istanza). Nel ruolo IAM, scegli il ruolo IAM che hai creato in precedenza. Completa la procedura come descritto.

> **Note**
>
> È necessario creare o utilizzare un gruppo di sicurezza e una coppia di chiavi esistenti per connettersi all'istanza.

Con questa configurazione di IAM e Amazon EC2, puoi distribuire la tua applicazione sull'istanza Amazon EC2 e avrà accesso in lettura al servizio Amazon S3.

Connect all'istanza EC2

Connect all'istanza EC2 in modo da poter trasferire l'applicazione di esempio su di essa e quindi eseguire l'applicazione. Avrai bisogno del file che contiene la parte privata della key pair che hai usato per avviare l'istanza, ovvero il file PEM.

Puoi farlo seguendo la procedura di connessione nella Guida per l'utente di Amazon EC2 o nella Guida per l'utente di Amazon EC2. Quando ti connetti, fallo in modo da poter trasferire i file dalla macchina di sviluppo all'istanza.

Se utilizzi un AWS Toolkit, spesso puoi connetterti all'istanza anche utilizzando il Toolkit. Per ulteriori informazioni, consulta la guida utente specifica per il Toolkit che utilizzi.

Esegui l'applicazione di esempio sull'istanza EC2

1. Copia i file dell'applicazione dall'unità locale all'istanza.

   Per informazioni su come trasferire file sulla tua istanza, consulta la Amazon EC2 User Guide o la Amazon EC2 User Guide.
2. Avvia l'applicazione e verifica che funzioni con gli stessi risultati della macchina di sviluppo.
3. (Facoltativo) Verifica che l'applicazione utilizzi le credenziali fornite dal ruolo IAM.

   a. Accedi AWS Management Console e apri la console Amazon EC2 all'indirizzo https://console.aws.amazon.com/ec2/.

   b. Seleziona l'istanza e scollega il ruolo IAM tramite Actions, Instance Settings, Attach/Replace IAM Role.

   c. Esegui nuovamente l'applicazione e conferma che restituisca un errore di autorizzazione.

Riferimento alle impostazioni

Gli SDK forniscono API specifiche per la lingua. Servizi AWS Si occupano di alcune delle attività più impegnative necessarie per effettuare correttamente le chiamate API, tra cui l'autenticazione, il comportamento dei tentativi di ripetizione e altro ancora. A tal fine, gli SDK dispongono di strategie flessibili per ottenere credenziali da utilizzare per le richieste, mantenere le impostazioni da utilizzare con ciascun servizio e ottenere valori da utilizzare per le impostazioni globali.

Puoi trovare informazioni dettagliate sulle impostazioni di configurazione nelle seguenti sezioni:

- AWS SDK e strumenti: fornitori di credenziali standardizzati— Provider di credenziali comuni standardizzati su più SDK.
- AWS SDK e strumenti: funzionalità standardizzate— Funzionalità comuni standardizzate su più SDK.

Creazione di client di servizio

Per accedere a livello di codice Servizi AWS, gli SDK utilizzano una classe/oggetto client per ciascuno. Servizio AWS Ad esempio, se l'applicazione deve accedere ad Amazon EC2, l'applicazione crea un oggetto client Amazon EC2 per interfacciarsi con quel servizio. Quindi utilizzi il client di servizio per effettuare richieste in merito. Servizio AWS Nella maggior parte degli SDK, un oggetto client di servizio è immutabile, quindi è necessario creare un nuovo client per ogni servizio a cui si effettuano richieste e per effettuare richieste allo stesso servizio utilizzando una configurazione diversa.

Precedenza delle impostazioni

Le impostazioni globali configurano funzionalità, fornitori di credenziali e altre funzionalità supportate dalla maggior parte degli SDK e che hanno un ampio impatto su tutti. Servizi AWS Tutti gli SDK hanno una serie di posizioni (o fonti) che controllano per trovare un valore per le impostazioni globali. Di seguito è riportata la priorità delle impostazioni di ricerca:

1. Qualsiasi impostazione esplicita impostata nel codice o su un client di servizio stesso ha la precedenza su qualsiasi altra cosa.
   - Alcune impostazioni possono essere impostate in base all'operazione e possono essere modificate secondo necessità per ogni operazione richiamata. Per l' AWS CLI operazione

AWS Tools for PowerShell, queste assumono la forma di parametri specifici per operazione che vengono immessi nella riga di comando. Per un SDK, le assegnazioni esplicite possono assumere la forma di un parametro impostato quando si crea un'istanza di un Servizio AWS client o di un oggetto di configurazione o, a volte, quando si chiama una singola API.

2. Solo Java/Kotlin: la proprietà del sistema JVM per l'impostazione è verificata. Se è impostato, quel valore viene utilizzato per configurare il client.
3. La variabile di ambiente è selezionata. Se è impostato, quel valore viene utilizzato per configurare il client.
4. L'SDK verifica l'impostazione nel `credentials` file condiviso. Se è impostato, il client lo utilizza.
5. Il `config` file condiviso per l'impostazione. Se l'impostazione è presente, l'SDK la utilizza.
   - La variabile di `AWS_PROFILE` ambiente o la proprietà di sistema `aws.profile` JVM possono essere utilizzate per specificare il profilo caricato dall'SDK.
6. Qualsiasi valore predefinito fornito dal codice sorgente SDK stesso viene utilizzato per ultimo.

> **Note**
>
> Alcuni SDK e strumenti potrebbero eseguire il check-in in un ordine diverso. Inoltre, alcuni SDK e strumenti supportano altri metodi di archiviazione e recupero dei parametri. Ad esempio, AWS SDK for .NET supporta una fonte aggiuntiva chiamata SDK Store. Per ulteriori informazioni sui provider esclusivi di un SDK o di uno strumento, consulta la guida specifica per l'SDK o lo strumento che stai utilizzando.

L'ordine determina quali metodi hanno la precedenza e sostituiscono gli altri. Ad esempio, se configuri un profilo nel `config` file condiviso, questo viene trovato e utilizzato solo dopo che l'SDK o lo strumento hanno prima verificato le altre posizioni. Ciò significa che se inserisci un'impostazione nel `credentials` file, questa viene utilizzata al posto di quella trovata nel `config` file. Se si configura una variabile di ambiente con un'impostazione e un valore, questa avrà la precedenza su tale impostazione `credentials` sia nei `config` file che. Infine, un'impostazione sulla singola operazione (parametro della AWS CLI riga di comando o parametro API) o nel codice sovrascriverebbe tutti gli altri valori di quell'unico comando.

`Config`elenco delle impostazioni dei file

Le impostazioni elencate nella tabella seguente possono essere assegnate nel AWS `config` file condiviso. Sono globali e riguardano tutti Servizi AWS. Gli SDK e gli strumenti possono anche supportare impostazioni e variabili di ambiente uniche. Per visualizzare le impostazioni e le variabili di ambiente supportate solo da un singolo SDK o strumento, consulta l'SDK o la guida agli strumenti specifici.

Nome dell'impostazione	Informazioni
`api_versions`	Impostazioni generali di configurazione
`aws_access_key_id`	AWS chiavi di accesso
`aws_secret_access_key`	AWS chiavi di accesso
`aws_session_token`	AWS chiavi di accesso
`ca_bundle`	impostazioni generali di configurazione
`credential_process`	Provider di credenziali di processo
`credential_source`	Assumi il ruolo di fornitore di credenziali
`defaults_mode`	Impostazioni predefinite di configurazione intelligente
`disable_request_compression`	Richiedi la compressione
`duration_seconds`	Assumi il ruolo di fornitore di credenziali

Nome dell'impostazione	Informazioni
`ec2_metadata_service_endpoint`	Fornitore di credenziali IMDS
`ec2_metadata_service_endpoint_mode`	Fornitore di credenziali IMDS
`ec2_metadata_v1_disabled`	Fornitore di credenziali IMDS
`endpoint_discovery_enabled`	Individuazione degli endpoint
`endpoint_url`	Endpoint specifici del servizio
`external_id`	Assumi il ruolo di fornitore di credenziali
`ignore_configured_endpoint_urls`	Endpoint specifici del servizio
`max_attempts`	Comportamento dei nuovi tentativi
`metadata_service_num_attempts`	Metadati delle istanze Amazon EC2
`metadata_service_timeout`	Metadati delle istanze Amazon EC2
`mfa_serial`	Assumi il ruolo di fornitore di credenziali

Nome dell'impostazione	Informazioni	
output	Impostazioni generali di configurazione	
parameter_validation	Impostazioni generali di configurazione	
region	Regione AWS	
request_min_compression_size_bytes	Richiedi la compressione	
retry_mode	Comportamento dei nuovi tentativi	
role_arn	Assumi il ruolo di fornitore di credenziali	
role_session_name	Assumi il ruolo di fornitore di credenziali	
s3_disable_multiregion_access_points	Punti di accesso multi-Regione di Amazon S3	
s3_use_arn_region	Access point Amazon S3	
sdk_ua_app_id	ID dell'applicazione	
source_profile	Assumi il ruolo di fornitore di credenziali	
sso_account_id	Provider di credenziali IAM Identity Center	
sso_region	Provider di credenziali IAM Identity Center	
sso_registration_scopes	Provider di credenziali IAM Identity Center	

Nome dell'impostazione	Informazioni	
`sso_role_name`	Provider di credenziali IAM Identity Center	
`sso_start_url`	Provider di credenziali IAM Identity Center	
`sts_regional_endpoints`	AWS STS Endpoint regionalizzati	
`use_dualstack_endpoint`	Endpoint dual-stack e FIPS	
`use_fips_endpoint`	Endpoint dual-stack e FIPS	
`web_identity_token_file`	Assumi il ruolo di fornitore di credenziali	

`Credentials`elenco delle impostazioni dei file

Le impostazioni elencate nella tabella seguente possono essere assegnate nel AWS `credentials` file condiviso. Sono globali e riguardano tutti Servizi AWS. Gli SDK e gli strumenti possono anche supportare impostazioni e variabili di ambiente uniche. Per visualizzare le impostazioni e le variabili di ambiente supportate solo da un singolo SDK o strumento, consulta l'SDK o la guida agli strumenti specifici.

Nome dell'impostazione	Informazioni	
`aws_access_key_id`	AWS chiavi di accesso	
`aws_secret_access_key`	AWS chiavi di accesso	
`aws_session_token`	AWS chiavi di accesso	

elenco delle variabili di ambiente

Le variabili di ambiente supportate dalla maggior parte degli SDK sono elencate nella tabella seguente. Sono globali e riguardano tutti Servizi AWS. Gli SDK e gli strumenti possono anche supportare impostazioni e variabili di ambiente uniche. Per visualizzare le impostazioni e le variabili di ambiente supportate solo da un singolo SDK o strumento, consulta l'SDK o la guida agli strumenti specifici.

Nome dell'impostazione	Informazioni	
`AWS_ACCESS_KEY_ID`	AWS chiavi di accesso	
`AWS_CA_BUNDLE`	impostazioni generali di configurazione	
`AWS_CONFIG_FILE`	Ubicazione degli elementi condivisi `config` e `credentials` dei file	
`AWS_CONTAINER_AUTHORIZATION_TOKEN`	Fornitore di credenziali per contenitori	
`AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE`	Fornitore di credenziali per contenitori	
`AWS_CONTAINER_CREDENTIALS_FULL_URI`	Fornitore di credenziali per contenitori	
`AWS_CONTAINER_CREDENTIALS_RELATIVE_URI`	Fornitore di credenziali per contenitori	

Nome dell'impostazione	Informazioni	
AWS_DEFAULTS_MODE	Impostazioni predefinite di configurazione intelligenti	
AWS_DISABLE_REQUEST_COMPRESSION	Richiedi la compressione	
AWS_EC2_METADATA_DISABLED	Fornitore di credenziali IMDS	
AWS_EC2_METADATA_SERVICE_ENDPOINT	Fornitore di credenziali IMDS	
AWS_EC2_METADATA_SERVICE_ENDPOINT_MODE	Fornitore di credenziali IMDS	
AWS_EC2_METADATA_V1_DISABLED	Fornitore di credenziali IMDS	
AWS_ENABLE_ENDPOINT_DISCOVERY	Individuazione degli endpoint	
AWS_ENDPOINT_URL	Endpoint specifici del servizio	
AWS_ENDPOINT_URL_<SERVICE>	Endpoint specifici del servizio	

Nome dell'impostazione	Informazioni
AWS_IAM_ROLE_ARN	Assumi il ruolo di fornitore di credenziali
AWS_IAM_ROLE_SESSION_NAME	Assumi il ruolo di fornitore di credenziali
AWS_IGNORE_CONFIGURED_ENDPOINT_URLS	Endpoint specifici del servizio
AWS_MAX_ATTEMPTS	Comportamento dei nuovi tentativi
AWS_METADATA_SERVICE_NUM_ATTEMPTS	Metadati delle istanze Amazon EC2
AWS_METADATA_SERVICE_TIMEOUT	Metadati delle istanze Amazon EC2
AWS_PROFILE	Condivisi `config` e file `credentials`
AWS_REGION	Regione AWS
AWS_REQUEST_MIN_COMPRESSION_SIZE_BYTES	Richiedi la compressione
AWS_RETRY_MODE	Comportamento dei nuovi tentativi

Nome dell'impostazione	Informazioni
`AWS_S3_DISABLE_MULTIREGION_ACCESS_POINTS`	Punti di accesso multi-Regione di Amazon S3
`AWS_S3_USE_ARN_REGION`	Access point Amazon S3
`AWS_SDK_UA_APP_ID`	ID dell'applicazione
`AWS_SECRET_ACCESS_KEY`	AWS chiavi di accesso
`AWS_SESSION_TOKEN`	AWS chiavi di accesso
`AWS_SHARED_CREDENTIALS_FILE`	Ubicazione dei `credentials` file condivisi `config` e
`AWS_STS_REGIONAL_ENDPOINTS`	AWS STS Endpoint regionalizzati
`AWS_USE_DUALSTACK_ENDPOINT`	Endpoint dual-stack e FIPS
`AWS_USE_FIPS_ENDPOINT`	Endpoint dual-stack e FIPS
`AWS_WEB_IDENTITY_TOKEN_FILE`	Assumi il ruolo di fornitore di credenziali

Elenco delle proprietà del sistema JVM

È possibile utilizzare le seguenti proprietà del sistema JVM per AWS SDK for Java e SDK AWS for Kotlin (destinate alla JVM). the section called “Come impostare le proprietà del sistema JVM”Per istruzioni su come impostare le proprietà del sistema JVM, vedere.

Nome dell'impostazione	Informazioni	
`aws.accessKeyId`	AWS chiavi di accesso	
`aws.configFile`	Ubicazione dei `credentials` file condivisi `config` e	
`aws.defaultsMode`	Impostazioni predefinite di configurazione intelligente	
`aws.disableEc2MetadataV1`	Fornitore di credenziali IMDS	
`aws.disableRequestCompression`	Richiedi la compressione	
`aws.ec2MetadataServiceEndpoint`	Fornitore di credenziali IMDS	
`aws.ec2MetadataServiceEndpointMode`	Fornitore di credenziali IMDS	
`aws.endpointDiscoveryEnabled`	Individuazione degli endpoint	
`aws.endpointUrl`	Endpoint specifici del servizio	

Nome dell'impostazione	Informazioni	
`aws.endpointUrl<ServiceName>`	Endpoint specifici del servizio	
`aws.ignoreConfiguredEndpointUrls`	Endpoint specifici del servizio	
`aws.maxAttempts`	Comportamento dei nuovi tentativi	
`aws.profile`	Condivisi `config` e file `credentials`	
`aws.region`	Regione AWS	
`aws.requestMinCompressionSizeBytes`	Richiedi la compressione	
`aws.retryMode`	Comportamento dei nuovi tentativi	
`aws.roleArn`	Assumi il ruolo di fornitore di credenziali	
`aws.roleSessionName`	Assumi il ruolo di fornitore di credenziali	
`aws.s3DisableMultiRegionAccessPoints`	Punti di accesso multi-Regione di Amazon S3	
`aws.s3UseArnRegion`	Access point Amazon S3	
`aws.secretAccessKey`	AWS chiavi di accesso	

Nome dell'impostazione	Informazioni	
`aws.sessionToken`	AWS chiavi di accesso	
`aws.sharedCredentialsFile`	Ubicazione dei `credentials` file condivisi `config` e	
`aws.useDualstackEndpoint`	Endpoint dual-stack e FIPS	
`aws.useFipsEndpoint`	Endpoint dual-stack e FIPS	
`aws.userAgentAppId`	ID dell'applicazione	
`aws.webIdentityTokenFile`	Assumi il ruolo di fornitore di credenziali	

AWS SDK e strumenti: fornitori di credenziali standardizzati

Molti fornitori di credenziali sono stati standardizzati in base a impostazioni predefinite coerenti e funzionano allo stesso modo su molti SDK. Questa coerenza aumenta la produttività e la chiarezza durante la codifica su più SDK. Tutte le impostazioni possono essere sovrascritte nel codice. Per i dettagli, consulta la tua API SDK specifica.

> **Important**
>
> Non tutti gli SDK supportano tutti i provider e nemmeno tutti gli aspetti all'interno di un provider.

Argomenti

- Catena di fornitori di credenziali
- AWS chiavi di accesso
- Assumi il ruolo di fornitore di credenziali
- Provider di credenziali per container
- Provider di credenziali IAM Identity Center
- Provider di credenziali IMDS
- Provider di credenziali di processo

Catena di fornitori di credenziali

Tutti gli SDK dispongono di una serie di posizioni (o fonti) che controllano per trovare credenziali valide da utilizzare per effettuare una richiesta a un. Servizio AWS Dopo aver trovato credenziali valide, la ricerca viene interrotta. Questa ricerca sistematica è chiamata catena di fornitori di credenziali predefinita.

Sebbene la catena distinta utilizzata da ciascun SDK sia diversa, il più delle volte include fonti come le seguenti:

Fornitore di credenziali	Descrizione
AWS chiavi di accesso	AWS chiavi di accesso per un utente IAM (ad esempio`AWS_ACCESS_KEY_ID` , and`AWS_SECRET_ACCESS_KEY`).
Federazione con identità web o OpenID Connect- Assumi il ruolo di fornitore di credenziali	Accedi utilizzando un provider di identità (IdP) esterno noto, come Login with Amazon, Facebook, Google o qualsiasi altro IdP compatibile con OpenID Connect (OIDC). Assumi le autorizzazioni di un ruolo IAM utilizzando un token di identità web di (). AWS Security Token Service AWS STS
Provider di credenziali IAM Identity Center	Ottieni credenziali da. AWS IAM Identity Center

Fornitore di credenziali	Descrizione
Assumi il ruolo di fornitore di credenziali	Ottieni l'accesso ad altre risorse assumendo le autorizza zioni di un ruolo IAM. (Recupera e usa le credenziali temporanee per un ruolo).
Provider di credenziali per container	Credenziali Amazon Elastic Container Service (Amazon ECS) e Amazon Elastic Kubernetes Service (Amazon EKS). Il provider di credenziali del contenitore recupera le credenziali per l'applicazione containerizzata del cliente.
Provider di credenziali di processo	Fornitore di credenziali personalizzate. Ottieni le tue credenziali da una fonte o da un processo esterno, incluso IAM Roles Anywhere.
Provider di credenziali IMDS	Credenziali del profilo dell'istanza Amazon Elastic Compute Cloud (Amazon EC2). Associa un ruolo IAM a ciascuna delle tue istanze EC2. Le credenziali temporane e per quel ruolo vengono rese disponibili per il codice in esecuzione nell'istanza. Le credenziali vengono fornite tramite il servizio di metadati Amazon EC2.

Per ogni fase della catena, esistono diversi modi per assegnare i valori di impostazione. L'impostazione dei valori specificati nel codice ha sempre la precedenza. Tuttavia, ci sono anche Variabili di ambiente e ilCondivisi `config` e `credentials` file. Per ulteriori informazioni, consulta Precedenza delle impostazioni.

AWS chiavi di accesso

> **Warning**
>
> Per evitare rischi per la sicurezza, non utilizzare gli utenti IAM per l'autenticazione quando sviluppi software creato ad hoc o lavori con dati reali. Utilizza invece la federazione con un provider di identità come AWS IAM Identity Center.

AWS le chiavi di accesso per un utente IAM possono essere utilizzate come AWS credenziali. L' AWS SDK utilizza automaticamente queste AWS credenziali per firmare le richieste API AWS, in modo che i carichi di lavoro possano accedere alle AWS risorse e ai dati in modo sicuro e conveniente. Si consiglia di utilizzare sempre il in `aws_session_token` modo che le credenziali siano temporanee e non più valide dopo la scadenza. L'utilizzo di credenziali a lungo termine non è consigliato.

> **Note**
>
> Se non AWS è possibile aggiornare queste credenziali temporanee, è AWS possibile estenderne la validità in modo da non influire sui carichi di lavoro.

Il AWS `credentials` file condiviso è la posizione consigliata per l'archiviazione delle informazioni sulle credenziali perché si trova in modo sicuro all'esterno delle directory di origine dell'applicazione e separato dalle impostazioni specifiche dell'SDK del file condiviso. `config`

Per ulteriori informazioni sulle AWS credenziali e sull'utilizzo delle chiavi di accesso, consulta le credenziali di AWS sicurezza e la gestione delle chiavi di accesso per gli utenti IAM nella Guida per l'utente IAM.

Configura questa funzionalità utilizzando quanto segue:

`aws_access_key_id`- impostazione dei AWS **`config`** file condivisi, **`aws_access_key_id`**- impostazione condivisa dei AWS **`credentials`** file (metodo consigliato), **`AWS_ACCESS_KEY_ID`**- variabile d'ambiente, **`aws.accessKeyId`**- Proprietà del sistema JVM: solo Java/Kotlin

Specifica la chiave di AWS accesso utilizzata come parte delle credenziali per autenticare l'utente.

`aws_secret_access_key`- impostazione dei file condivisi AWS **`config`**, **`aws_secret_access_key`**- impostazione condivisa dei AWS **`credentials`** file (metodo consigliato), **`AWS_SECRET_ACCESS_KEY`**- variabile d'ambiente, **`aws.secretAccessKey`**- Proprietà del sistema JVM: solo Java/Kotlin

Specifica la chiave AWS segreta utilizzata come parte delle credenziali per autenticare l'utente.

`aws_session_token`- impostazione dei file condivisi AWS **`config`**, **`aws_session_token`**- impostazione condivisa dei AWS **`credentials`** file (metodo consigliato), **`AWS_SESSION_TOKEN`**- variabile d'ambiente, **`aws.sessionToken`**- Proprietà del sistema JVM: solo Java/Kotlin

> Specifica un token di AWS sessione utilizzato come parte delle credenziali per autenticare l'utente. Questo valore viene ricevuto come parte delle credenziali temporanee restituite dalle richieste di assunzione di un ruolo riuscite. È richiesto un token di sessione solo se si specificano manualmente credenziali di sicurezza temporanee. Tuttavia, ti consigliamo di utilizzare sempre credenziali di sicurezza temporanee anziché credenziali a lungo termine. Per consigli sulla sicurezza, consulta Best practice di sicurezza in IAM.

Per istruzioni su come ottenere questi valori, consultaAutenticazione utilizzando credenziali a breve termine.

Esempio di impostazione di questi valori obbligatori nel `credentials` file `config` or:

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

Esempio in Linux/macOS di impostazione delle variabili di ambiente tramite riga di comando:

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

Esempio in Windows di impostazione delle variabili di ambiente tramite riga di comando:

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

Compatibilità con gli AWS SDK

I seguenti SDK supportano le funzionalità e le impostazioni descritte in questo argomento. Vengono annotate eventuali eccezioni parziali. Tutte le impostazioni delle proprietà del sistema JVM sono supportate solo da AWS SDK for Java and the. SDK AWS for Kotlin

SDK	Supporto	Note o ulteriori informazioni
AWS CLI v2	Sì	
SDK per C++	Sì	`config`file condiviso non supportato.
SDK per Go V2 (1.x)	Sì	
SDK per Go 1.x (V1)	Sì	Per utilizzare le impostazioni dei `config` file condivisi, devi attivare il caricamento dal file di configurazione; vedi Sessioni.
SDK per Java 2.x	Sì	
SDK per Java 1.x	Sì	
SDK per 3.x JavaScript	Sì	
SDK per 2.x JavaScript	Sì	
SDK per Kotlin	Sì	
SDK per.NET 3.x	Sì	Variabili di ambiente non supportate.
SDK per PHP 3.x	Sì	
SDK per Python (Boto3)	Sì	
SDK per Ruby 3.x	Sì	
SDK per Rust	Sì	
Strumenti per PowerShell	Sì	Variabili di ambiente non supportate.

Assumi il ruolo di fornitore di credenziali

Assumere un ruolo implica l'utilizzo di un set di credenziali di sicurezza temporanee per accedere a AWS risorse a cui altrimenti non avresti accesso. Le credenziali temporanee sono costituite da un ID chiave di accesso, una chiave di accesso segreta e un token di sicurezza.

Per configurare l'SDK o lo strumento per assumere un ruolo, devi prima creare o identificare un ruolo specifico da assumere. I ruoli IAM sono identificati in modo univoco da un ruolo Amazon Resource Name (ARN). I ruoli stabiliscono relazioni di fiducia con un'altra entità. L'entità affidabile che utilizza il ruolo potrebbe essere un'altra Servizio AWS Account AWS, un provider di identità Web o una federazione OIDC o SAML.

Dopo aver identificato il ruolo IAM, se quel ruolo ti affida la fiducia, puoi configurare il tuo SDK o lo strumento per utilizzare le autorizzazioni concesse dal ruolo. A tale scopo, utilizza le seguenti impostazioni.

Per indicazioni su come iniziare a utilizzare queste impostazioni, Assunzione di un ruolo consulta questa guida.

Assumi le impostazioni del fornitore di credenziali di ruolo

Configura questa funzionalità utilizzando quanto segue:

`credential_source`- impostazione dei AWS **`config`** file condivisi

Utilizzato all'interno delle istanze Amazon EC2 o dei contenitori Amazon Elastic Container Service per specificare dove l'SDK o lo strumento possono trovare le credenziali autorizzate ad assumere il ruolo specificato con il parametro. `role_arn`

Valore predefinito: Nessuno

Valori validi:

- Ambiente: specifica che l'SDK o lo strumento devono recuperare le credenziali di origine dalle variabili di ambiente e. `AWS_ACCESS_KEY_IDAWS_SECRET_ACCESS_KEY`
- Ec2 InstanceMetadata: specifica che l'SDK o lo strumento deve utilizzare il ruolo IAM collegato al profilo dell'istanza EC2 per ottenere le credenziali di origine.
- EcsContainer— Speciifica che l'SDK o lo strumento deve utilizzare il ruolo IAM collegato al contenitore ECS per ottenere le credenziali di origine.

Non è possibile specificare sia `credential_source` sia `source_profile` nello stesso profilo.

Esempio di impostazione in un `config` file per indicare che le credenziali devono provenire da Amazon EC2:

```
credential_source = Ec2InstanceMetadata
```

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

`duration_seconds`- impostazione di file condivisi AWS **`config`**

Specifica la durata massima della sessione del ruolo, in secondi.

Questa impostazione si applica solo quando il profilo specifica di assumere un ruolo.

Valore predefinito: 3600 secondi (un'ora)

Valori validi: il valore può variare da 900 secondi (15 minuti) fino all'impostazione della durata massima della sessione configurata per il ruolo (che può essere un massimo di 43200 secondi o 12 ore). Per ulteriori informazioni, consulta Visualizza l'impostazione della durata massima della sessione per un ruolo nella Guida per l'utente IAM.

Esempio di impostazione di questa impostazione in un `config` file:

```
duration_seconds = 43200
```

`external_id`- impostazione di AWS **`config`** file condivisi

Specifica un identificatore univoco che viene utilizzato da terze parti per assumere un ruolo negli account dei relativi clienti.

Questa impostazione si applica solo quando il profilo specifica di assumere un ruolo e la politica di fiducia per il ruolo richiede un valore per`ExternalId`. Il valore è mappato al `ExternalId` parametro che viene passato all'`AssumeRole`operazione quando il profilo specifica un ruolo.

Valore predefinito: Nessuno.

Valori validi: vedi Come utilizzare un ID esterno per concedere l'accesso alle tue AWS risorse a una terza parte nella Guida per l'utente IAM.

Esempio di impostazione in un `config` file:

```
external_id = unique_value_assigned_by_3rd_party
```

`mfa_serial`- impostazione di AWS **`config`** file condivisi

Speciifica l'identificazione o il numero di serie di un dispositivo di autenticazione a più fattori (MFA) che l'utente deve utilizzare quando assume un ruolo.

Richiesto quando si assume un ruolo in cui la politica di attendibilità per quel ruolo include una condizione che richiede l'autenticazione MFA.

Valore predefinito: Nessuno.

Valori validi: il valore può essere un numero di serie per un dispositivo hardware (ad esempio`GAHT12345678`) o un Amazon Resource Name (ARN) per un dispositivo MFA virtuale. Per ulteriori informazioni sull'MFA, consulta Configurazione dell'accesso all'API protetto da MFA nella Guida per l'utente IAM.

Esempio di impostazione in un file: `config`

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

`role_arn`- impostazione di AWS **`config`** file condivisi, **`AWS_IAM_ROLE_ARN`**- variabile d'ambiente, **`aws.roleArn`**- Proprietà del sistema JVM: solo Java/Kotlin

Speciifica l'Amazon Resource Name (ARN) di un ruolo IAM che desideri utilizzare per eseguire le operazioni richieste utilizzando questo profilo.

Valore predefinito: Nessuno.

Valori validi: il valore deve essere l'ARN di un ruolo IAM, formattato come segue: `arn:aws:iam::account-id:role/role-name`

Inoltre, è necessario specificare anche una delle seguenti impostazioni:

- `source_profile`— Per identificare un altro profilo da utilizzare per trovare le credenziali autorizzate ad assumere il ruolo in questo profilo.
- `credential_source`— Utilizzare credenziali identificate dalle variabili di ambiente correnti o credenziali allegate a un profilo di istanza Amazon EC2 o a un'istanza di container Amazon ECS.
- `web_identity_token_file`— Utilizzare provider di identità pubblici o qualsiasi provider di identità compatibile con OpenID Connect (OIDC) per gli utenti che sono stati autenticati in un'applicazione mobile o web.

`role_session_name`- impostazione di file condivisi AWS **`config`**, **`AWS_IAM_ROLE_SESSION_NAME`**- variabile d'ambiente, **`aws.roleSessionName`**- Proprietà del sistema JVM: solo Java/Kotlin

Specifica il nome da associare alla sessione del ruolo. Questo nome appare nei AWS CloudTrail registri delle voci associate a questa sessione, il che può essere utile durante il controllo.

Valore predefinito: un parametro opzionale. Se non fornisci questo valore, viene generato automaticamente un nome di sessione se il profilo assume un ruolo.

Valori validi: forniti al `RoleSessionName` parametro quando l' AWS API AWS CLI o l'API richiama l'`AssumeRole`operazione (o operazioni come l'`AssumeRoleWithWebIdentity`operazione) per tuo conto. Il valore diventa parte dell'utente presunto Amazon Resource Name (ARN) che puoi interrogare e viene visualizzato come parte delle voci di CloudTrail registro per le operazioni richiamate da questo profilo.

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

Esempio di impostazione di questo valore in un `config` file:

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

`source_profile`- impostazione di AWS **`config`** file condivisi

Specifica un altro profilo le cui credenziali vengono utilizzate per assumere il ruolo specificato dall'`role_arn`impostazione nel profilo originale. Per informazioni su come vengono utilizzati i profili negli archivi condivisi AWS `config` e nei `credentials` file, consulta. Condivisi `config` e `credentials` file

Se si specifica un profilo che è anche un profilo di assunzione, ogni ruolo verrà assunto in ordine sequenziale per risolvere completamente le credenziali. Questa catena viene interrotta quando l'SDK incontra un profilo con credenziali. Il concatenamento dei ruoli limita la sessione di ruolo AWS CLI o dell' AWS API a un massimo di un'ora e non può essere aumentato. Per ulteriori informazioni, consulta i termini e i concetti relativi ai ruoli nella Guida per l'utente IAM.

Valore predefinito: Nessuno.

Valori validi: una stringa di testo costituita dal nome di un profilo definito nei `credentials` file `config` and. È inoltre necessario specificare un valore per `role_arn` nel profilo corrente.

Non è possibile specificare sia `credential_source` sia `source_profile` nello stesso profilo.

Esempio di impostazione in un file di configurazione:

```
[profile A]
source_profile = B
```

```
role_arn =  arn:aws:iam::123456789012:role/RoleA

[profile B]
aws_access_key_id=AKIAIOSFODNN7EXAMPLE
aws_secret_access_key=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
aws_session_token=IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZVERYLONGS
```

Nell'esempio precedente il profilo A utilizza le credenziali del profilo B. Quando specificate che l'SDK o lo strumento devono utilizzare il A profilo, l'SDK o lo strumento cerca automaticamente le credenziali per il B profilo collegato e le utilizza per richiedere credenziali temporanee per il ruolo IAM specificato. L'SDK o lo strumento utilizza l'operazione sts: in background per AssumeRole eseguire questa operazione. Queste credenziali temporanee vengono quindi utilizzate dal codice per accedere alle risorse. AWS Al ruolo specificato devono essere associate politiche di autorizzazione IAM che consentano l'esecuzione del codice richiesto, ad esempio il comando o il Servizio AWS metodo API.

`web_identity_token_file`- impostazione condivisa AWS **`config`** dei file, **`AWS_WEB_IDENTITY_TOKEN_FILE`**- variabile d'ambiente, **`aws.webIdentityTokenFile`**- Proprietà del sistema JVM: solo Java/Kotlin

Speciifica il percorso di un file che contiene un token di accesso da un provider OAuth 2.0 supportato o da un provider di identità OpenID Connect ID.

Questa impostazione consente l'autenticazione utilizzando provider di federazione delle identità Web, come Google, Facebook e Amazon, tra molti altri. L'SDK o lo strumento di sviluppo carica il contenuto di questo file e lo passa come `WebIdentityToken` argomento quando chiama l'`AssumeRoleWithWebIdentity`operazione per conto dell'utente.

Valore predefinito: Nessuno.

Valori validi: questo valore deve essere un percorso e un nome di file. Il file deve contenere un token di accesso OAuth 2.0 o un token OpenID Connect fornito da un provider di identità. I percorsi relativi vengono trattati come relativi alla directory di lavoro del processo.

Compatibilità con gli AWS SDK

I seguenti SDK supportano le funzionalità e le impostazioni descritte in questo argomento. Vengono annotate eventuali eccezioni parziali. Tutte le impostazioni delle proprietà del sistema JVM sono supportate solo da AWS SDK for Java and the. SDK AWS for Kotlin

SDK	Su o	Note o ulteriori informazioni
AWS CLI v2	Sì	
SDK per C++	Parzia	`credential_source` non supportato. `duration_seconds` non supportato. `mfa_serial` non supportato.
SDK per Go V2 (1.x)	Sì	
SDK per Go 1.x (V1)	Sì	Per utilizzare le impostazioni dei `config` file condivisi, devi attivare il caricamento dal file di configurazione; vedi Sessioni.
SDK per Java 2.x	Parzia	`mfa_serial` non supportato. Usa `AWS_ROLE_ARN` al posto di`AWS_IAM_ROLE_ARN` . Utilizzare `AWS_ROLE_SESSION_NAME` al posto di`AWS_IAM_ROLE_SESSION_NAME` .
SDK per Java 1.x	Parzia	`mfa_serial` non supportato. Le proprietà del sistema JVM non sono supportate.
SDK per 3.x JavaScript	Sì	
SDK per 2.x JavaScript	Parzia	`credential_source` non supportato.
SDK per Kotlin	Sì	Usa invece di`AWS_ROLE_ARN` . `AWS_IAM_ROLE_ARN` Utilizzare `AWS_ROLE_SESSION_NAME` al posto di`AWS_IAM_ROLE_SESSION_NAME` .
SDK per.NET 3.x	Sì	
SDK per PHP 3.x	Sì	
SDK per Python (Boto3)	Sì	
SDK per Ruby 3.x	Sì	
SDK per Rust	Sì	
Strumenti per PowerShell	Sì	

Provider di credenziali per container

Il provider di credenziali del contenitore recupera le credenziali per l'applicazione containerizzata del cliente. Questo provider di credenziali è utile per i clienti di Amazon Elastic Container Service (Amazon ECS) e Amazon Elastic Kubernetes Service (Amazon EKS). Gli SDK tentano di caricare le credenziali dall'endpoint HTTP specificato tramite una richiesta GET.

Se utilizzi Amazon ECS, ti consigliamo di utilizzare un task IAM Role per migliorare l'isolamento, l'autorizzazione e la verificabilità delle credenziali. Una volta configurato, Amazon ECS imposta la variabile di `AWS_CONTAINER_CREDENTIALS_RELATIVE_URI` ambiente utilizzata dagli SDK e dagli strumenti per ottenere le credenziali. Per configurare Amazon ECS per questa funzionalità, consulta il ruolo Task IAM nella Amazon Elastic Container Service Developer Guide.

Se usi Amazon EKS, ti consigliamo di utilizzare Amazon EKS Pod Identity per migliorare l'isolamento delle credenziali, il privilegio minimo, la verificabilità, il funzionamento indipendente, la riusabilità e la scalabilità. Sia il tuo ruolo Pod che un ruolo IAM sono associati a un account di servizio Kubernetes per gestire le credenziali per le tue applicazioni. Per ulteriori informazioni su Amazon EKS Pod Identity, consulta Amazon EKS Pod Identities nella Amazon EKS User Guide. Una volta configurato, Amazon EKS imposta le `AWS_CONTAINER_CREDENTIALS_FULL_URI` variabili di `AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE` ambiente che gli SDK e gli strumenti utilizzano per ottenere le credenziali. Per informazioni sulla configurazione, consulta Configurazione dell'agente Amazon EKS Pod Identity nella Guida per l'utente di Amazon EKS o Amazon EKS Pod Identity semplifica le autorizzazioni IAM per le applicazioni sui cluster Amazon EKS sul sito Web del AWS blog.

Configura questa funzionalità utilizzando quanto segue:

`AWS_CONTAINER_CREDENTIALS_FULL_URI`- variabile di ambiente

Specifica l'endpoint URL HTTP completo per l'SDK da utilizzare quando si effettua una richiesta di credenziali. Ciò include sia lo schema che l'host.

Valore predefinito: Nessuno.

Valori validi: URI valido.

Nota: questa impostazione è un'alternativa *`AWS_CONTAINER_CREDENTIALS_RELATIVE_URI`* e verrà utilizzata solo se non *`AWS_CONTAINER_CREDENTIALS_RELATIVE_URI`* è impostata.

Esempio in Linux/macOS di impostazione delle variabili di ambiente tramite riga di comando:

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

oppure

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

`AWS_CONTAINER_CREDENTIALS_RELATIVE_URI`- variabile di ambiente

Specifica l'endpoint URL HTTP relativo per l'SDK da utilizzare quando si effettua una richiesta di credenziali. Il valore viene aggiunto al nome host Amazon ECS predefinito di. `169.254.170.2`

Valore predefinito: Nessuno.

Valori validi: URI relativo valido.

Esempio in Linux/macOS di impostazione delle variabili di ambiente tramite riga di comando:

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

`AWS_CONTAINER_AUTHORIZATION_TOKEN`- variabile di ambiente

Specifica un token di autorizzazione in testo semplice. Se questa variabile è impostata, l'SDK imposterà l'intestazione Authorization sulla richiesta HTTP con il valore della variabile di ambiente.

Valore predefinito: Nessuno.

Valori validi: String.

Nota: questa impostazione è un'alternativa *`AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE`* e verrà utilizzata solo se non *`AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE`* è impostata.

Esempio in Linux/macOS di impostazione delle variabili di ambiente tramite riga di comando:

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

`AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE`- variabile di ambiente

Speciifica il percorso assoluto di un file che contiene il token di autorizzazione in testo semplice.

Valore predefinito: Nessuno.

Valori validi: String.

Esempio in Linux/macOS di impostazione delle variabili di ambiente tramite riga di comando:

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

Compatibilità con AWS gli SDK

I seguenti SDK supportano le funzionalità e le impostazioni descritte in questo argomento. Vengono annotate eventuali eccezioni parziali. Tutte le impostazioni delle proprietà del sistema JVM sono supportate solo da AWS SDK for Java and the. SDK AWS for Kotlin

SDK	Supporto	Note o ulteriori informazioni
AWS CLI v2	Sì	
SDK per C++	Sì	
SDK per Go V2 (1.x)	Sì	
SDK per Go 1.x (V1)	Sì	
SDK per Java 2.x	Sì	
SDK per Java 1.x	Parziale	Amazon EKS Pod Identity e `AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE` non sono supportati.
SDK per 3.x JavaScript	Sì	
SDK per 2.x JavaScript	Sì	
SDK per Kotlin	Sì	
SDK per.NET 3.x	Sì	
SDK per PHP 3.x	Sì	
SDK per Python (Boto3)	Sì	
SDK per Ruby 3.x	Sì	

SDK	Su o	Note o ulteriori informazioni
SDK per Rust	Sì	
Strumenti per PowerShell	Sì	

Provider di credenziali IAM Identity Center

Questo meccanismo di autenticazione consente AWS IAM Identity Center di ottenere l'accesso Single Sign-On (SSO) al codice. Servizi AWS

> **Note**
>
> Nella documentazione dell'API AWS SDK, il provider di credenziali IAM Identity Center è chiamato provider di credenziali SSO.

Dopo aver abilitato IAM Identity Center, definisci un profilo per le relative impostazioni nel file condiviso. AWS `config` Questo profilo viene utilizzato per connettersi al portale di accesso IAM Identity Center. Quando un utente si autentica con successo con IAM Identity Center, il portale restituisce credenziali a breve termine per il ruolo IAM associato a quell'utente. Per scoprire come l'SDK ottiene credenziali temporanee dalla configurazione e le utilizza per le richieste, consulta. Servizio AWS Comprendi l'autenticazione IAM Identity Center

Esistono due modi per configurare IAM Identity Center tramite il `config` file:

- Configurazione del provider di token SSO (consigliata): durate di sessione estese.
- Configurazione legacy non aggiornabile: utilizza una sessione fissa di otto ore.

In entrambe le configurazioni, è necessario accedere nuovamente alla scadenza della sessione.

Per impostare durate di sessione personalizzate, è necessario utilizzare la configurazione del provider di token SSO.

Le due guide seguenti contengono informazioni aggiuntive su IAM Identity Center:

- AWS IAM Identity Center Guida per l'utente

- AWS IAM Identity Center Riferimento all'API del portale

Prerequisiti

Devi prima abilitare IAM Identity Center. Per i dettagli sull'attivazione dell'autenticazione IAM Identity Center, consulta la Guida introduttiva nella Guida AWS IAM Identity Center per l'utente.

In alternativa, segui le Autenticazione IAM Identity Center istruzioni contenute in questa guida. Queste istruzioni forniscono una guida completa, dall'attivazione di IAM Identity Center al completamento della necessaria configurazione dei `config` file condivisi che segue qui.

Configurazione del provider di token SSO

> **Note**
>
> Per utilizzare AWS CLI per creare questa configurazione per te, vedi Configurare il tuo profilo con la `aws configure sso` procedura guidata in. AWS CLI

Quando utilizzi la configurazione del provider di token SSO, l' AWS SDK o lo strumento aggiorna automaticamente la sessione fino al periodo di sessione prolungato. Per ulteriori informazioni sulla durata della sessione e sulla durata massima, consulta Configurare la durata della sessione del portale di AWS accesso e delle applicazioni integrate IAM Identity Center nella Guida per l'AWS IAM Identity Center utente.

La `sso-session` sezione del `config` file viene utilizzata per raggruppare le variabili di configurazione per l'acquisizione dei token di accesso SSO, che possono quindi essere utilizzati per acquisire le credenziali. AWS Per ulteriori dettagli sulla formattazione delle sezioni all'interno di un file, vedere. `config` Formato del file di configurazione

Si definisce una `sso-session` sezione e la si associa a un profilo. `sso_regione` `sso_start_url` deve essere impostato all'interno della `sso-session` sezione. In genere, `sso_account_id` e `sso_role_name` deve essere impostato nella `profile` sezione in modo che l'SDK possa richiedere AWS le credenziali.

> **Note**
>
> Per un'analisi approfondita su come gli SDK e gli strumenti utilizzano e aggiornano le credenziali utilizzando questa configurazione, consulta. Comprendi l'autenticazione IAM Identity Center

L'esempio seguente configura l'SDK per richiedere le credenziali IAM Identity Center. Supporta anche l'aggiornamento automatico dei token.

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

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

È possibile riutilizzare le `sso-session` configurazioni su più profili.

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

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

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

`sso_account_id` e `sso_role_name` non sono necessari per tutti gli scenari di configurazione del token SSO. Se l'applicazione utilizza solo il supporto per Servizi AWS l'autenticazione al portatore, non sono necessarie AWS le credenziali tradizionali. L'autenticazione Bearer è uno schema di

autenticazione HTTP che utilizza token di sicurezza chiamati bearer tokens. In questo scenario, `sso_account_id` e `sso_role_name` non sono obbligatori. Consulta la guida individuale per Servizio AWS determinare se supporta l'autorizzazione del token al portatore.

Gli ambiti di registrazione sono configurati come parte di un. `sso-session` L'ambito è un meccanismo OAuth 2.0 per limitare l'accesso di un'applicazione all'account di un utente. Un'applicazione può richiedere uno o più ambiti e il token di accesso rilasciato all'applicazione è limitato agli ambiti concessi. Questi ambiti definiscono le autorizzazioni richieste per l'autorizzazione per il client OIDC registrato e i token di accesso recuperati dal client. Per le opzioni di ambito di accesso supportate, consulta Ambiti di accesso nella Guida per l'utente.AWS IAM Identity Center L'esempio seguente fornisce l'accesso per `sso_registration_scopes` elencare account e ruoli.

```
[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
```

Il token di autenticazione viene memorizzato nella cache su disco all'interno della `~/.aws/sso/cache` directory con un nome di file basato sul nome della sessione.

Configurazione legacy non aggiornabile

L'aggiornamento automatico dei token non è supportato utilizzando la configurazione legacy non aggiornabile. Ti consigliamo invece di utilizzare. Configurazione del provider di token SSO

Per utilizzare la configurazione precedente non aggiornabile, è necessario specificare le seguenti impostazioni all'interno del profilo:

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

Si specifica il portale utente per un profilo con le impostazioni `sso_start_url` and`sso_region`. Le autorizzazioni vengono specificate con le `sso_role_name` impostazioni `sso_account_id` e.

L'esempio seguente imposta i quattro valori obbligatori nel `config` file.

```
[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
```

Il token di autenticazione viene memorizzato nella cache su disco all'interno della `~/.aws/sso/cache` directory con un nome di file basato su. `sso_start_url`

Impostazioni del provider di credenziali IAM Identity Center

Configura questa funzionalità utilizzando quanto segue:

`sso_start_url`- impostazione dei AWS **`config`** file condivisi

L'URL che rimanda al portale di accesso IAM Identity Center della tua organizzazione. Per ulteriori informazioni sul portale di accesso IAM Identity Center, consulta Using the AWS access portal nella AWS IAM Identity Center User Guide.

Per trovare questo valore, apri la console IAM Identity Center, visualizza la dashboard e trova AWS l'URL del portale di accesso.

`sso_region`- impostazione dei AWS **`config`** file condivisi

Il Regione AWS che contiene l'host del portale IAM Identity Center, ovvero la regione selezionata prima di abilitare IAM Identity Center. È indipendente dalla AWS regione predefinita e può essere diversa.

Per un elenco completo di Regioni AWS e dei relativi codici, consulta Endpoint regionali nel Riferimenti generali di Amazon Web Services. Per trovare questo valore, apri la console IAM Identity Center, visualizza la dashboard e trova Region.

`sso_account_id`- impostazione dei AWS **`config`** file condivisi

L'ID numerico di Account AWS che è stato aggiunto tramite il AWS Organizations servizio da utilizzare per l'autenticazione.

Per visualizzare l'elenco degli account disponibili, vai alla console IAM Identity Center e apri la Account AWSpagina. Puoi anche visualizzare l'elenco degli account disponibili utilizzando il metodo ListAccountsAPI nel AWS IAM Identity Center Portal API Reference. Ad esempio, puoi chiamare il AWS CLI metodo list-accounts.

`sso_role_name`- impostazione di file condivisi AWS **`config`**

Il nome di un set di autorizzazioni fornito come ruolo IAM che definisce le autorizzazioni risultanti dell'utente. Il ruolo deve esistere nel file Account AWS specificato da. `sso_account_id` Usa il nome del ruolo, non il ruolo Amazon Resource Name (ARN).

I set di autorizzazioni sono associati a politiche IAM e politiche di autorizzazione personalizzate e definiscono il livello di accesso degli utenti ai dati loro assegnati. Account AWS

Per visualizzare l'elenco dei set di autorizzazioni disponibili per Account AWS, vai alla console IAM Identity Center e apri la Account AWSpagina. Scegli il nome corretto del set di autorizzazioni elencato nella Account AWS tabella. Puoi anche visualizzare l'elenco dei set di autorizzazioni disponibili utilizzando il metodo ListAccountRolesAPI nel AWS IAM Identity Center Portal API Reference. Ad esempio, puoi chiamare il AWS CLI metodo list-account-roles.

`sso_registration_scopes`- impostazione di AWS **`config`** file condivisi

Un elenco delimitato da virgole di stringhe di ambito valide da autorizzare per. `sso-session` Gli ambiti autorizzano l'accesso agli endpoint autorizzati con token portatore di IAM Identity Center. È `sso:account:access` necessario concedere un ambito minimo di per ottenere un token di aggiornamento dal servizio IAM Identity Center. Per le stringhe di ambito di accesso supportate, consulta Access scopes nella Guida per l'AWS IAM Identity Center utente. Questa impostazione non si applica alla configurazione precedente non aggiornabile. I token emessi utilizzando la configurazione legacy sono limitati all'ambito implicito. `sso:account:access`

Compatibilità con gli SDK AWS

I seguenti SDK supportano le funzionalità e le impostazioni descritte in questo argomento. Vengono annotate eventuali eccezioni parziali. Tutte le impostazioni delle proprietà del sistema JVM sono supportate solo da AWS SDK for Java and the. SDK AWS for Kotlin

SDK	Su o	Note o ulteriori informazioni
AWS CLI v2	Sì	
SDK per C++	Sì	
SDK per Go V2 (1.x)	Sì	

SDK	Su o	Note o ulteriori informazioni
SDK per Go 1.x (V1)	Sì	Per utilizzare le impostazioni dei `config` file condivisi, devi attivare il caricamento dal file di configurazione; vedi Sessioni.
SDK per Java 2.x	Sì	Valori di configurazione supportati anche nel `credentials` file.
SDK per Java 1.x	No	
SDK per 3.x JavaScript	Sì	
SDK per 2.x JavaScript	Sì	
SDK per Kotlin	Sì	
SDK per.NET 3.x	Sì	
SDK per PHP 3.x	Sì	
SDK per Python (Boto3)	Sì	
SDK per Ruby 3.x	Sì	
SDK per Rust	Parzia	Solo configurazione precedente non aggiornabile.
Strumenti per PowerShell	Sì	

Provider di credenziali IMDS

Instance Metadata Service (IMDS) fornisce dati sull'istanza che puoi utilizzare per configurare o gestire l'istanza in esecuzione. Per ulteriori informazioni sui dati disponibili, consulta Metadati e dati utente dell'istanza nella Amazon EC2 User Guide o Metadati e dati utente dell'istanza nella Amazon EC2 User Guide. Amazon EC2 fornisce un endpoint locale disponibile per le istanze in grado di fornire vari bit di informazioni all'istanza. Se all'istanza è associato un ruolo, può fornire un set di credenziali valide per quel ruolo. Gli SDK possono utilizzare quell'endpoint per risolvere le credenziali come parte della catena di provider di credenziali predefinita. Per impostazione predefinita, viene utilizzata la versione 2 di Instance Metadata Service (IMDSv2), una versione più sicura di IMDS che

utilizza un token di sessione. Se ciò non riesce a causa di una condizione non riutilizzabile (codici di errore HTTP 403, 404, 405), IMDSv1 viene utilizzato come fallback.

Configura questa funzionalità utilizzando quanto segue:

`AWS_EC2_METADATA_DISABLED`- variabile di ambiente

Se tentare o meno di utilizzare Amazon EC2 Instance Metadata Service (IMDS) per ottenere le credenziali.

Valore predefinito: `false`.

Valori validi:

- **`true`**— Non utilizzare IMDS per ottenere credenziali.
- **`false`**— Usa IMDS per ottenere le credenziali.

`ec2_metadata_v1_disabled`- impostazione di file condivisi AWS **`config`**, **`AWS_EC2_METADATA_V1_DISABLED`**- variabile d'ambiente, **`aws.disableEc2MetadataV1`**- Proprietà del sistema JVM: solo Java/Kotlin

Se utilizzare o meno Instance Metadata Service Version 1 (IMDSv1) come fallback in caso di errore di IMDSv2.

> **Note**
>
> I nuovi SDK non supportano IMDSv1 e, pertanto, non supportano questa impostazione. Per i dettagli, consulta la tabella. Compatibilità con gli SDK AWS

Valore predefinito: `false`.

Valori validi:

- **`true`**— Non utilizzare IMDSv1 come fallback.
- **`false`**— Usa IMDSv1 come fallback.

`ec2_metadata_service_endpoint`- impostazione condivisa dei file AWS **`config`**, **`AWS_EC2_METADATA_SERVICE_ENDPOINT`**- variabile d'ambiente, **`aws.ec2MetadataServiceEndpoint`**- Proprietà del sistema JVM: solo Java/Kotlin

L'endpoint di IMDS.

Valore predefinito: se è `ec2_metadata_service_endpoint_mode` ugualeIPv4, l'endpoint predefinito è. `http://169.254.169.254` Se è `ec2_metadata_service_endpoint_mode` uguale, l'endpoint predefinito è. `IPv6` `http://[fd00:ec2::254]`

Valori validi: URI valido.

`ec2_metadata_service_endpoint_mode`- impostazione di AWS **`config`** file condivisi, **`AWS_EC2_METADATA_SERVICE_ENDPOINT_MODE`**- variabile d'ambiente, **`aws.ec2MetadataServiceEndpointMode`**- Proprietà del sistema JVM: solo Java/Kotlin

La modalità endpoint di IMDS.

Valore predefinito:. `IPv4`

Valori validi:`IPv4`,`IPv6`.

> **Note**
>
> Il provider di credenziali IMDS fa parte di. Catena di fornitori di credenziali Tuttavia, il fornitore di credenziali IMDS viene controllato solo dopo diversi altri provider di questa serie. Pertanto, se si desidera che il programma utilizzi le credenziali di questo provider, è necessario rimuovere altri provider di credenziali validi dalla configurazione o utilizzare un profilo diverso. In alternativa, anziché affidarsi alla catena di fornitori di credenziali per scoprire automaticamente quale provider restituisce credenziali valide, specifica l'uso del provider di credenziali IMDS nel codice. È possibile specificare le fonti delle credenziali direttamente quando si creano client di servizio.

Sicurezza per le credenziali IMDS

Per impostazione predefinita, quando l' AWS SDK non è configurato con credenziali valide, l'SDK tenterà di utilizzare Amazon EC2 Instance Metadata Service (IMDS) per recuperare le credenziali per un ruolo. AWS Questo comportamento può essere disabilitato impostando la variabile di ambiente su. `AWS_EC2_METADATA_DISABLED` `true` Ciò impedisce attività di rete non necessarie e migliora la sicurezza su reti non attendibili in cui l'Amazon EC2 Instance Metadata Service può essere impersonato.

> **Note**
>
> AWS I client SDK configurati con credenziali valide non utilizzeranno mai IMDS per recuperare le credenziali, indipendentemente da nessuna di queste impostazioni.

Disabilitazione dell'uso delle credenziali IMDS di Amazon EC2

Il modo in cui imposti questa variabile di ambiente dipende dal sistema operativo in uso e dal fatto che desideri o meno che la modifica sia persistente.

Linux e macOS

I clienti che utilizzano Linux o macOS possono impostare questa variabile di ambiente con il seguente comando:

```
$ export AWS_EC2_METADATA_DISABLED=true
```

Se desideri che questa impostazione sia persistente tra più sessioni di shell e riavvii del sistema, puoi aggiungere il comando precedente al file del profilo della shell, ad esempio `.bash_profile`.`zsh_profile`, o. `.profile`

Windows

I clienti che utilizzano Windows possono impostare questa variabile di ambiente con il seguente comando:

```
$ set AWS_EC2_METADATA_DISABLED=true
```

Se desideri che questa impostazione sia persistente tra più sessioni di shell e riavvii del sistema, puoi utilizzare invece il seguente comando:

```
$ setx AWS_EC2_METADATA_DISABLED=true
```

> **Note**
>
> Il setx comando non applica il valore alla sessione di shell corrente, quindi sarà necessario ricaricare o riaprire la shell affinché la modifica abbia effetto.

Compatibilità con gli SDK AWS

I seguenti SDK supportano le funzionalità e le impostazioni descritte in questo argomento. Vengono annotate eventuali eccezioni parziali. Tutte le impostazioni delle proprietà del sistema JVM sono supportate solo da AWS SDK for Java and the. SDK AWS for Kotlin

SDK	Supporto	Note o ulteriori informazioni
AWS CLI v2	Sì	
SDK per C++	Sì	
SDK per Go V2 (1.x)	Sì	
SDK per Go 1.x (V1)	Sì	Per utilizzare le impostazioni dei `config` file condivisi, devi attivare il caricamento dal file di configurazione; vedi Sessioni.
SDK per Java 2.x	Sì	
SDK per Java 1.x	Parziale	Proprietà del sistema JVM: utilizzate `com.amazonaws.sdk.disableEc2MetadataV1` al posto di `aws.disableEc2MetadataV1` ; `aws.ec2MetadataServiceEndpoint` e non supportate. `aws.ec2MetadataServiceEndpointMode`
SDK per 3.x JavaScript	Sì	
SDK per 2.x JavaScript	Sì	
SDK per Kotlin	Sì	Non utilizza il fallback IMDSv1.
SDK per.NET 3.x	Sì	
SDK per PHP 3.x	Sì	
SDK per Python (Boto3)	Sì	
SDK per Ruby 3.x	Sì	
SDK per Rust	Sì	Non utilizza il fallback IMDSv1.

SDK	Su o	Note o ulteriori informazioni
Strumenti per PowerShell	Sì	È possibile disabilitare il fallback IMDSv1 in modo esplicito nel codice utilizzando. `[Amazon.Util.EC2InstanceMetadata]::EC2MetadataV1Disabled = $true`

Provider di credenziali di processo

Gli SDK forniscono un modo per estendere la catena di fornitori di credenziali per casi d'uso personalizzati.

IAM Roles Anywhere offre un modo per ottenere credenziali temporanee per un carico di lavoro o un processo eseguito all'esterno di. AWS Per configurarlo `credential_process` per questo utilizzo, consulta. IAM Roles Anywhere

> **Warning**
>
> Di seguito viene descritto un metodo di acquisizione delle credenziali da un processo esterno. Questo può essere potenzialmente pericoloso, quindi procedi con cautela. Se possibile, dovrebbero essere preferiti altri fornitori di credenziali. Se si utilizza questa opzione, è necessario assicurarsi che il `config` file sia il più protetto possibile utilizzando le migliori pratiche di sicurezza per il sistema operativo in uso. Verifica che lo strumento per le credenziali personalizzato non scriva informazioni segreteStdErr, poiché gli SDK AWS CLI possono acquisire e registrare tali informazioni, esponendole potenzialmente a utenti non autorizzati.

Configura questa funzionalità utilizzando quanto segue:

`credential_process`- impostazione dei AWS **`config`** file condivisi

Specifica un comando esterno che l'SDK o lo strumento esegue per conto dell'utente per generare o recuperare le credenziali di autenticazione da utilizzare. L'impostazione specifica il nome di un programma/comando che verrà richiamato dall'SDK. Quando l'SDK richiama il processo, attende che il processo scriva i dati JSON. `stdout` Il provider personalizzato deve restituire le informazioni in un formato specifico. Tali informazioni contengono le credenziali che l'SDK o lo strumento possono utilizzare per autenticare l'utente.

> **Note**
>
> Il provider delle credenziali di processo fa parte di. Catena di fornitori di credenziali Tuttavia, il fornitore delle credenziali di processo viene controllato solo dopo diversi altri provider di questa serie. Pertanto, se si desidera che il programma utilizzi le credenziali di questo provider, è necessario rimuovere altri provider di credenziali validi dalla configurazione o utilizzare un profilo diverso. In alternativa, anziché affidarsi alla catena di fornitori di credenziali per scoprire automaticamente quale provider restituisce credenziali valide, specificate l'uso del provider di credenziali di processo nel codice. È possibile specificare le fonti delle credenziali direttamente quando si creano client di servizio.

Specificare il percorso del programma di credenziali

Il valore dell'impostazione è una stringa che contiene il percorso di un programma che l'SDK o lo strumento di sviluppo esegue per conto dell'utente:

- Il percorso e il nome del file possono essere composti solo dai seguenti caratteri: A-Z, a-z, 0-9, trattino (-), trattino basso (_), punto (.), barra (/), barra rovesciata (\) e spazio.
- Se il percorso o il nome del file contiene uno spazio, circondare il percorso completo e il nome del file con virgolette doppie (" ").
- Se un nome di parametro o un valore di parametro contiene uno spazio, circondare tale elemento con virgolette doppie (" "). È possibile racchiudere solo il nome o il valore, non la coppia.
- Non includere alcuna variabile di ambiente nelle stringhe. Ad esempio, non includere `$HOME` o`%USERPROFILE%`.
- Non specificare la cartella home come~. * È necessario specificare il percorso completo o il nome del file di base. Se è presente un nome di file di base, il sistema tenta di trovare il programma all'interno delle cartelle specificate dalla variabile di `PATH` ambiente.

  L'esempio seguente mostra l'impostazione di credential_process nel file condiviso `config` su Linux/macOS.

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

  L'esempio seguente mostra l'impostazione di credential_process nel file condiviso su Windows. `config`

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

Output valido dal programma di credenziali

L'SDK esegue il comando come specificato nel profilo e quindi legge i dati dal flusso di output standard. Il comando specificato, che si tratti di uno script o di un programma binario, deve generare un output JSON STD0UT che corrisponda alla sintassi seguente.

```
{
    "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**
>
> Al momento della stesura del presente documento, la chiave `Version` deve essere configurata su 1. Questo valore potrebbe incrementare nel tempo, man mano che la struttura evolve.

La `Expiration` chiave è un timestamp in formato RFC3339. Se la `Expiration` chiave non è presente nell'output dello strumento, l'SDK presuppone che le credenziali siano credenziali a lungo termine che non si aggiornano. Altrimenti, le credenziali sono considerate credenziali temporanee e vengono aggiornate automaticamente eseguendo nuovamente il comando prima della scadenza delle credenziali. `credential_process`

> **Note**
>
> L'SDK non memorizza nella cache le credenziali dei processi esterni nello stesso modo in cui utilizza le credenziali di ruolo. Se il caching è necessario, dovrai implementarlo nel processo esterno.

Il processo esterno può restituire un codice diverso da zero per indicare che si è verificato un errore durante il recupero delle credenziali.

AWS Compatibilità con gli SDK

I seguenti SDK supportano le funzionalità e le impostazioni descritte in questo argomento. Vengono annotate eventuali eccezioni parziali. Tutte le impostazioni delle proprietà del sistema JVM sono supportate solo da AWS SDK for Java and the. SDK AWS for Kotlin

SDK	Su o	Note o ulteriori informazioni
AWS CLI v2	Sì	
SDK per C++	Sì	
SDK per Go V2 (1.x)	Sì	
SDK per Go 1.x (V1)	Sì	Per utilizzare le impostazioni dei `config` file condivisi, devi attivare il caricamento dal file di configurazione; vedi Sessioni.
SDK per Java 2.x	Sì	
SDK per Java 1.x	Sì	
SDK per 3.x JavaScript	Sì	
SDK per 2.x JavaScript	Sì	
SDK per Kotlin	Sì	
SDK per.NET 3.x	Sì	
SDK per PHP 3.x	Sì	
SDK per Python (Boto3)	Sì	
SDK per Ruby 3.x	Sì	
SDK per Rust	Sì	

SDK	Su o	Note o ulteriori informazioni
Strumenti per PowerShell	Sì	

AWS SDK e strumenti: funzionalità standardizzate

Molte funzionalità sono state standardizzate in base a impostazioni predefinite coerenti e per funzionare allo stesso modo su molti SDK. Questa coerenza aumenta la produttività e la chiarezza durante la codifica su più SDK. Tutte le impostazioni possono essere sovrascritte nel codice, consulta la tua API SDK specifica per i dettagli.

> **Important**
>
> Non tutti gli SDK supportano tutte le funzionalità o anche tutti gli aspetti all'interno di una funzionalità.

Argomenti

- ID dell'applicazione
- Metadati delle istanze Amazon EC2.
- Punti di accesso Amazon S3
- Punti di accesso multi-Regione di Amazon S3
- Regione AWS
- AWS STS Endpoint regionalizzati
- Endpoint dual-stack e FIPS
- Rilevamento di endpoint
- Impostazioni generali di configurazione
- Cliente IMDS
- Comportamento di ripetizione
- Richiesta di compressione
- Endpoint specifici del servizio

- Impostazioni predefinite di configurazione intelligente

ID dell'applicazione

Una singola Account AWS può essere utilizzata da più applicazioni del cliente a cui effettuare Servizi AWS chiamate. L'ID dell'applicazione consente ai clienti di identificare quale applicazione di origine ha effettuato una serie di chiamate utilizzando un Account AWS. AWS Gli SDK e i servizi non utilizzano o interpretano questo valore se non per renderlo visibile nelle comunicazioni con i clienti. Ad esempio, questo valore può essere incluso nelle e-mail operative o nel file AWS Health Dashboard per identificare in modo univoco quale delle applicazioni è associata alla notifica.

Configura questa funzionalità utilizzando quanto segue:

`sdk_ua_app_id`- impostazione dei AWS **`config`** file condivisi, **`AWS_SDK_UA_APP_ID`**- variabile d'ambiente, **`aws.userAgentAppId`**- Proprietà del sistema JVM: solo Java/Kotlin

> Questa impostazione è una stringa unica che assegnate alla vostra applicazione per identificare a quale delle vostre applicazioni all'interno di un particolare particolare effettua chiamate. Account AWS AWS
>
> Valore predefinito: `None`
>
> Valori validi: stringa con lunghezza massima di 50. Sono consentite lettere, numeri e i seguenti caratteri speciali: `!$,%,&,*,+,-,.,,,^,_,`,|,~`.

Esempio di impostazione di questo valore nel `config` file:

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

Esempio in Linux/macOS di impostazione delle variabili di ambiente tramite riga di comando:

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

Esempio in Windows di impostazione delle variabili di ambiente tramite riga di comando:

```
setx AWS_SDK_UA_APP_ID ABCDEF
```

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

Se includete simboli che hanno un significato speciale per la shell utilizzata, evitate il valore appropriato.

Compatibilità con gli AWS SDK

I seguenti SDK supportano le funzionalità e le impostazioni descritte in questo argomento. Vengono annotate eventuali eccezioni parziali. Tutte le impostazioni delle proprietà del sistema JVM sono supportate solo da AWS SDK for Java and the. SDK AWS for Kotlin

SDK	Su o	Note o ulteriori informazioni
AWS CLI v2	No	
SDK per C++	Sì	`config`file condiviso non supportato.
SDK per Go V2 (1.x)	Sì	
SDK per Go 1.x (V1)	No	
SDK per Java 2.x	Parzia	L'impostazione dei `config` file condivisi non è supportata; la variabile di ambiente non è supportata.
SDK per Java 1.x	No	
SDK per 3.x JavaScript	Sì	
SDK per 2.x JavaScript	No	
SDK per Kotlin	Sì	
SDK per.NET 3.x	Sì	Variabili di ambiente non supportate.
SDK per PHP 3.x	Sì	
SDK per Python (Boto3)	Sì	
SDK per Ruby 3.x	Sì	

SDK	Su o	Note o ulteriori informazioni
SDK per Rust	Sì	
Strumenti per PowerShell	No	

Metadati delle istanze Amazon EC2.

Amazon EC2 fornisce un servizio su istanze chiamato Instance Metadata Service (IMDS). Per ulteriori informazioni su questo servizio, consulta Metadati e dati utente dell'istanza nella Amazon EC2 User Guide o Metadati e dati utente dell'istanza nella Amazon EC2 User Guide. Quando si tenta di recuperare le credenziali su un'istanza Amazon EC2 configurata con un ruolo IAM, la connessione al servizio di metadati dell'istanza è regolabile.

Configura questa funzionalità utilizzando quanto segue:

`metadata_service_num_attempts`- impostazione dei AWS **`config`** file condivisi, **`AWS_METADATA_SERVICE_NUM_ATTEMPTS`**- variabile d'ambiente

Questa impostazione specifica il numero totale di tentativi da effettuare prima di rinunciare al tentativo di recuperare dati dal servizio di metadati dell'istanza.

Valore predefinito: 1

Valori validi: numero maggiore o uguale a 1.

`metadata_service_timeout`- impostazione condivisa AWS **`config`** dei file, **`AWS_METADATA_SERVICE_TIMEOUT`**- variabile d'ambiente

Speciifica il numero di secondi prima del timeout quando si tenta di recuperare i dati dal servizio di metadati dell'istanza.

Valore predefinito: 1

Valori validi: numero maggiore o uguale a 1.

Esempio di impostazione di questi valori nel `config` file:

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

```
metadata_service_timeout=10
```

Esempio in Linux/macOS di impostazione delle variabili di ambiente tramite riga di comando:

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

Esempio in Windows di impostazione delle variabili di ambiente tramite riga di comando:

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

Compatibilità con gli AWS SDK

I seguenti SDK supportano le funzionalità e le impostazioni descritte in questo argomento. Vengono annotate eventuali eccezioni parziali. Tutte le impostazioni delle proprietà del sistema JVM sono supportate solo da AWS SDK for Java and the. SDK AWS for Kotlin

SDK	Supporto	Note o ulteriori informazioni
AWS CLI v2	Sì	
SDK per C++	No	
SDK per Go V2 (1.x)	No	
SDK per Go 1.x (V1)	No	
SDK per Java 2.x	No	
SDK per Java 1.x	Parziale	Solo `AWS_METADATA_SERVICE_TIMEOUT` è supportata.
SDK per 3.x JavaScript	No	
SDK per 2.x JavaScript	No	
SDK per Kotlin	No	
SDK per.NET 3.x	No	

SDK	Su o	Note o ulteriori informazioni
SDK per PHP 3.x	Sì	
SDK per Python (Boto3)	Sì	
SDK per Ruby 3.x	No	
SDK per Rust	No	
Strumenti per PowerShell	No	

Punti di accesso Amazon S3

Il servizio Amazon S3 fornisce punti di accesso come modo alternativo per interagire con i bucket Amazon S3. Gli access point hanno politiche e configurazioni uniche che possono essere applicate a loro anziché direttamente al bucket. Con AWS gli SDK, puoi utilizzare il punto di accesso Amazon Resource Names (ARN) nel campo del bucket per le operazioni API invece di specificare il nome del bucket in modo esplicito. Vengono utilizzati per operazioni specifiche come l'utilizzo di un punto di accesso ARN `GetObject`per recuperare un oggetto da un bucket o l'utilizzo di un punto di accesso ARN `PutObject`per aggiungere un oggetto a un bucket.

Per ulteriori informazioni sui punti di accesso e gli ARN di Amazon S3, consulta Using access point nella Amazon S3 User Guide.

Configura questa funzionalità utilizzando quanto segue:

`s3_use_arn_region`- impostazione dei AWS **`config`** file condivisi, **`AWS_S3_USE_ARN_REGION`**- variabile d'ambiente, **`aws.s3UseArnRegion`**- Proprietà del sistema JVM: solo Java/Kotlin, Per configurare il valore direttamente nel codice, consulta direttamente il tuo SDK specifico.

> Questa impostazione controlla se l'SDK utilizza l' Regione AWS ARN del punto di accesso per costruire l'endpoint regionale per la richiesta. L'SDK verifica che l'ARN Regione AWS sia servito dalla stessa AWS partizione configurata dal client Regione AWS per evitare chiamate tra partizioni che molto probabilmente falliranno. Se definita in modo multiplo, l'impostazione configurata dal codice ha la precedenza, seguita dall'impostazione della variabile di ambiente.
>
> Valore predefinito: `false`

Valori validi:

- **`true`**— L'SDK utilizza gli ARN Regione AWS durante la costruzione dell'endpoint anziché quelli configurati dal client. Regione AWS Eccezione: se la configurazione del client Regione AWS è un FIPS Regione AWS, deve corrispondere a quella dell'ARN. Regione AWS In caso contrario verrà restituito un errore.
- **`false`**— L'SDK utilizza la configurazione del client Regione AWS durante la costruzione dell'endpoint.

Compatibilità con gli SDK AWS

I seguenti SDK supportano le funzionalità e le impostazioni descritte in questo argomento. Vengono annotate eventuali eccezioni parziali. Tutte le impostazioni delle proprietà del sistema JVM sono supportate solo da AWS SDK for Java and the. SDK AWS for Kotlin

SDK	Su o	Note o ulteriori informazioni
AWS CLI v2	Sì	
SDK per C++	Sì	
SDK per Go V2 (1.x)	Sì	
SDK per Go 1.x (V1)	Sì	Per utilizzare le impostazioni dei `config` file condivisi, devi attivare il caricamento dal file di configurazione; vedi Sessioni.
SDK per Java 2.x	Sì	
SDK per Java 1.x	Sì	Proprietà di sistema JVM non supportata.
SDK per 3.x JavaScript	Sì	
SDK per 2.x JavaScript	Sì	
SDK per Kotlin	Sì	
SDK per.NET 3.x	Sì	Non segue la precedenza standard; il valore del `config` file condiviso ha la precedenza sulla variabile di ambiente.

SDK	Su o	Note o ulteriori informazioni
SDK per PHP 3.x	Sì	
SDK per Python (Boto3)	Sì	
SDK per Ruby 3.x	Sì	
SDK per Rust	No	
Strumenti per PowerShell	Sì	Non segue la precedenza standard; il valore `config` del file condiviso ha la precedenza sulla variabile di ambiente.

Punti di accesso multi-Regione di Amazon S3

Gli access point multiregionali di Amazon S3 forniscono un endpoint globale che le applicazioni possono utilizzare per soddisfare le richieste provenienti da bucket Amazon S3 distribuiti in più aree. Regioni AWSÈ possibile utilizzare punti di accesso multiregionali per creare applicazioni multiregionali con la stessa architettura utilizzata in una singola regione e quindi eseguire tali applicazioni in qualsiasi parte del mondo.

Per ulteriori informazioni sui punti di accesso multiregionali, consulta Punti di accesso multiregionali in Amazon S3 nella Guida per l'utente di Amazon S3.

Per ulteriori informazioni sui nomi di risorse Amazon Resource Names (ARN) per punti di accesso multiregionali, consulta Effettuare richieste utilizzando un punto di accesso multiregionale nella Guida per l'utente di Amazon S3.

Per ulteriori informazioni sulla creazione di punti di accesso multiregionali, consulta Managing Multi-Region Access Points nella Amazon S3 User Guide.

L'algoritmo SigV4A è l'implementazione di firma utilizzata per firmare le richieste regionali globali. Questo algoritmo è ottenuto dall'SDK tramite una dipendenza da. AWSLibrerie Common Runtime (CRT)

Configura questa funzionalità utilizzando quanto segue:

`s3_disable_multiregion_access_points`- impostazione dei AWS **`config`** file condivisi, **`AWS_S3_DISABLE_MULTIREGION_ACCESS_POINTS`**- variabile d'ambiente, **`aws.s3DisableMultiRegionAccessPoints`**- Proprietà del sistema JVM: solo Java/Kotlin, Per configurare il valore direttamente nel codice, consulta direttamente il tuo SDK specifico.

Questa impostazione controlla se l'SDK tenta potenzialmente di effettuare richieste interregionali. Se definita in modo multiplo, l'impostazione configurata dal codice ha la precedenza, seguita dall'impostazione della variabile di ambiente.

Valore predefinito: `false`

Valori validi:

- **`true`**— Interrompe l'uso delle richieste interregionali.
- **`false`**— Abilita le richieste interregionali utilizzando punti di accesso multiregionali.

Compatibilità con gli SDK AWS

I seguenti SDK supportano le funzionalità e le impostazioni descritte in questo argomento. Vengono annotate eventuali eccezioni parziali. Tutte le impostazioni delle proprietà del sistema JVM sono supportate solo da AWS SDK for Java and the. SDK AWS for Kotlin

SDK	Su o	Note o ulteriori informazioni
AWS CLI v2	Sì	
SDK per C++	Sì	
SDK per Go V2 (1.x)	Sì	
SDK per Go 1.x (V1)	No	
SDK per Java 2.x	Sì	
SDK per Java 1.x	No	
SDK per 3.x JavaScript	Sì	
SDK per 2.x JavaScript	No	

SDK	Su o	Note o ulteriori informazioni
SDK per Kotlin	Sì	
SDK per.NET 3.x	Sì	
SDK per PHP 3.x	Sì	
SDK per Python (Boto3)	Sì	
SDK per Ruby 3.x	Sì	
SDK per Rust	Sì	
Strumenti per PowerShell	Sì	

Regione AWS

Regioni AWS sono un concetto importante da comprendere quando si lavora Servizi AWS.

Con Regioni AWS, puoi accedere a chi Servizi AWS risiede fisicamente in un'area geografica specifica. Ciò può essere utile per mantenere attivi i dati e le applicazioni in prossimità del luogo in cui voi e i vostri utenti potrete accedervi. Le regioni forniscono la tolleranza ai guasti, la stabilità e la resilienza e possono anche ridurre la latenza. Con Regions, puoi creare risorse ridondanti che rimangono disponibili e non sono interessate da un'interruzione regionale.

La maggior parte delle Servizio AWS richieste è associata a una particolare area geografica. Le risorse create in una regione non esistono in nessun'altra regione a meno che non si utilizzi esplicitamente una funzionalità di replica offerta da un. Servizio AWS Ad esempio, Amazon S3 e Amazon EC2 supportano la replica tra regioni. Alcuni servizi, come IAM, non dispongono di risorse regionali.

Riferimenti generali di AWSContiene informazioni su quanto segue:

- Per comprendere la relazione tra le regioni e gli endpoint e per visualizzare un elenco degli endpoint regionali esistenti, consulta AWS Service Endpoint.
- Per visualizzare l'elenco corrente di tutte le regioni e gli endpoint supportati per ciascuna Servizio AWS, consulta Endpoint e quote del servizio.

Creazione di client di servizio

Per accedere a livello di codice Servizi AWS, gli SDK utilizzano una classe/oggetto client per ciascuno. Servizio AWS Se l'applicazione deve accedere ad Amazon EC2, ad esempio, l'applicazione creerà un oggetto client Amazon EC2 per interfacciarsi con quel servizio.

Se non viene specificata in modo esplicito alcuna regione per il client, per impostazione predefinita il client utilizza la regione impostata tramite l'impostazione seguente. `region` Tuttavia, la regione attiva per un client può essere impostata in modo esplicito per ogni singolo oggetto client. L'impostazione della Regione in questo modo ha la precedenza su qualsiasi impostazione globale per quel particolare client di servizio. La regione alternativa viene specificata durante la creazione di un'istanza di quel client, specifica per il tuo SDK (consulta la guida SDK specifica o la base di codice dell'SDK).

Configura questa funzionalità utilizzando quanto segue:

`region`- impostazione dei AWS **`config`** file condivisi, **`AWS_REGION`**- variabile d'ambiente, **`aws.region`**- Proprietà del sistema JVM: solo Java/Kotlin

Specifica l'impostazione predefinita da utilizzare per le richieste. Regione AWS AWS Questa regione viene utilizzata per le richieste di servizio SDK a cui non viene fornita una regione specifica da utilizzare.

Valore predefinito: Nessuno. È necessario specificare questo valore in modo esplicito.

Valori validi:

- Tutti i codici regionali disponibili per il servizio scelto, elencati negli endpoint del AWS servizio nel Riferimento AWS generale. Ad esempio, il valore `us-east-1` imposta l'endpoint sugli Regione AWS Stati Uniti orientali (Virginia settentrionale).
- `aws-global`specifica l'endpoint globale per i servizi che supportano un endpoint globale separato oltre agli endpoint regionali, come AWS Security Token Service ()AWS STS e Amazon Simple Storage Service (Amazon S3).

Esempio di impostazione di questo valore nel file: `config`

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

Esempio in Linux/macOS di impostazione delle variabili di ambiente tramite riga di comando:

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

Esempio in Windows di impostazione delle variabili di ambiente tramite riga di comando:

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

La maggior parte degli SDK dispone di un oggetto di «configurazione» che è disponibile per impostare la regione predefinita all'interno del codice dell'applicazione. Per i dettagli, consulta la tua guida per sviluppatori AWS SDK specifica.

Compatibilità con AWS gli SDK

I seguenti SDK supportano le funzionalità e le impostazioni descritte in questo argomento. Vengono annotate eventuali eccezioni parziali. Tutte le impostazioni delle proprietà del sistema JVM sono supportate solo da AWS SDK for Java and the. SDK AWS for Kotlin

SDK	Supporto	Note o ulteriori informazioni
AWS CLI v2	Sì	AWS CLI v2 utilizza qualsiasi valore `AWS_REGION` prima di qualsiasi valore in `AWS_DEFAULT_REGION` (entrambe le variabili vengono controllate).
AWS CLI v1	Sì	AWS CLI v1 utilizza una variabile di ambiente denominata a questo `AWS_DEFAULT_REGION` scopo.
SDK per C++	Sì	
SDK per Go V2 (1.x)	Sì	
SDK per Go 1.x (V1)	Sì	Per utilizzare le impostazioni dei `config` file condivisi, devi attivare il caricamento dal file di configurazione; vedi Sessioni.
SDK per Java 2.x	Sì	
SDK per Java 1.x	Sì	
SDK per 3.x JavaScript	Sì	

SDK	Su o	Note o ulteriori informazioni
SDK per 2.x JavaScript	Sì	
SDK per Kotlin	Sì	
SDK per.NET 3.x	Sì	
SDK per PHP 3.x	Sì	
SDK per Python (Boto3)	Sì	Questo SDK utilizza una variabile di ambiente denominata a questo scopo. `AWS_DEFAULT_REGION`
SDK per Ruby 3.x	Sì	
SDK per Rust	Sì	
Strumenti per PowerShell	Sì	

AWS STS Endpoint regionalizzati

Per impostazione predefinita, AWS Security Token Service (AWS STS) è disponibile come servizio globale e tutte le AWS STS richieste vanno a un singolo endpoint all'indirizzo. `https://sts.amazonaws.com` Le richieste globali si riferiscono alla regione Stati Uniti orientali (Virginia settentrionale). AWS consiglia di utilizzare gli AWS STS endpoint regionali anziché l'endpoint globale. Per ulteriori informazioni sugli AWS STS endpoint, consulta Endpoints nell'API Reference.AWS Security Token Service

Configura questa funzionalità utilizzando quanto segue:

`sts_regional_endpoints`- impostazione dei AWS **`config`** file condivisi,
`AWS_STS_REGIONAL_ENDPOINTS`- variabile d'ambiente

> Questa impostazione specifica in che modo l'SDK o lo strumento determina l' Servizio AWS endpoint che utilizza per comunicare con (). AWS Security Token Service AWS STS
>
> Valore predefinito: `legacy`

> **Note**
>
> Tutte le nuove versioni principali dell'SDK rilasciate dopo luglio 2022 verranno utilizzate per impostazione predefinita. `regional` Le nuove versioni principali dell'SDK potrebbero rimuovere questa impostazione e questo comportamento d'uso. `regional` Per ridurre l'impatto futuro di questa modifica, ti consigliamo di iniziare a `regional` utilizzarla nella tua applicazione quando possibile.

Valori validi: (Valore consigliato:*`regional`*)

- **`legacy`**— Utilizza l' AWS STS endpoint global`sts.amazonaws.com`, per le seguenti AWS regioni:`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`, e`us-west-2`. Tutte le altre regioni utilizzano automaticamente il rispettivo endpoint regionale.
- **`regional`**— L'SDK o lo strumento utilizza sempre l' AWS STS endpoint per la regione attualmente configurata. Ad esempio, se il client è configurato per l'uso`us-west-2`, tutte le chiamate AWS STS vengono effettuate all'endpoint regionale`sts.us-west-2.amazonaws.com`, anziché all'endpoint globale. `sts.amazonaws.com` Per inviare una richiesta all'endpoint globale mentre questa impostazione è abilitata, è possibile impostare l'area geografica su `aws-global`.

Esempio di impostazione di questi valori nel `config` file:

```
[default]
sts_regional_endpoints = regional
```

Esempio in Linux/macOS di impostazione delle variabili di ambiente tramite riga di comando:

```
export AWS_STS_REGIONAL_ENDPOINTS=regional
```

Esempio in Windows di impostazione delle variabili di ambiente tramite riga di comando:

```
setx AWS_STS_REGIONAL_ENDPOINTS regional
```

Compatibilità con gli AWS SDK

I seguenti SDK supportano le funzionalità e le impostazioni descritte in questo argomento. Vengono annotate eventuali eccezioni parziali. Tutte le impostazioni delle proprietà del sistema JVM sono supportate solo da AWS SDK for Java and the. SDK AWS for Kotlin

SDK	Su o	Note o ulteriori informazioni
AWS CLI v2	Parzia	Il valore predefinito è `regional`.
SDK per C++	Parzia	Variabile di ambiente e impostazione `config` dei file non supportate. L'SDK funziona con l'`regional`impostazione.
SDK per Go V2 (1.x)	Sì	
SDK per Go 1.x (V1)	Sì	Per utilizzare le impostazioni dei `config` file condivisi, devi attivare il caricamento dal file di configurazione; vedi Sessioni.
SDK per Java 2.x	Sì	
SDK per Java 1.x	Sì	
SDK per 3.x JavaScript	Sì	
SDK per 2.x JavaScript	Sì	
SDK per Kotlin	No	
SDK per.NET 3.x	Sì	
SDK per PHP 3.x	Sì	
SDK per Python (Boto3)	Sì	
SDK per Ruby 3.x	Sì	
SDK per Rust	Sì	
Strumenti per PowerShell	Sì	

Endpoint dual-stack e FIPS

Configura questa funzionalità utilizzando quanto segue:

`use_dualstack_endpoint`- impostazione dei AWS **`config`** file condivisi, **`AWS_USE_DUALSTACK_ENDPOINT`**- variabile d'ambiente, **`aws.useDualstackEndpoint`**- Proprietà del sistema JVM: solo Java/Kotlin

Attiva o disattiva se l'SDK invierà richieste agli endpoint dual-stack. Per ulteriori informazioni sugli endpoint dual-stack, che supportano sia il traffico IPv4 che IPv6, consulta Using Using Amazon S3 dual-stack endpoint nella Amazon Simple Storage Service User Guide. Gli endpoint dual-stack sono disponibili per alcuni servizi in alcune regioni.

Valore predefinito: `false`

Valori validi:

- **`true`**— L'SDK o lo strumento tenteranno di utilizzare gli endpoint dual-stack per effettuare richieste di rete. Se non esiste un endpoint dual-stack per il servizio e/o, la richiesta avrà esito negativo. Regione AWS
- **`false`**— L'SDK o lo strumento non utilizzeranno endpoint dual-stack per effettuare richieste di rete.

`use_fips_endpoint`- impostazione di AWS **`config`** file condivisi, **`AWS_USE_FIPS_ENDPOINT`**- variabile d'ambiente, **`aws.useFipsEndpoint`**- Proprietà del sistema JVM: solo Java/Kotlin

Attiva o disattiva se l'SDK o lo strumento invieranno richieste agli endpoint conformi a FIPS. I Federal Information Processing Standards (FIPS) sono un insieme di requisiti di sicurezza del governo degli Stati Uniti per i dati e la loro crittografia. Le agenzie governative, i partner e coloro che desiderano fare affari con il governo federale sono tenuti a rispettare le linee guida FIPS. A differenza degli AWS endpoint standard, gli endpoint FIPS utilizzano una libreria software TLS conforme a FIPS 140-2. Se questa impostazione è abilitata e non esiste un endpoint FIPS per il servizio in uso, la chiamata potrebbe non riuscire. Regione AWS AWS Endpoint specifici del servizioe l'`--endpoint-url`opzione per AWS Command Line Interface sovrascrivere questa impostazione.

Per ulteriori informazioni su altri modi per specificare gli endpoint FIPS tramite Regione AWS, consulta FIPS Endpoints by Service. Per ulteriori informazioni sugli endpoint del servizio Amazon Elastic Compute Cloud, consulta gli endpoint Dual-stack (IPv4 e IPv6) nell'Amazon EC2 API Reference.

Valore predefinito: `false`

Valori validi:

- **`true`**— L'SDK o lo strumento invieranno le richieste agli endpoint conformi a FIPS.
- **`false`**— L'SDK o lo strumento non invieranno richieste agli endpoint conformi a FIPS.

Compatibilità con gli AWS SDK

I seguenti SDK supportano le funzionalità e le impostazioni descritte in questo argomento. Vengono annotate eventuali eccezioni parziali. Tutte le impostazioni delle proprietà del sistema JVM sono supportate solo da AWS SDK for Java and the. SDK AWS for Kotlin

SDK	Su o	Note o ulteriori informazioni
AWS CLI v2	Sì	
SDK per C++	Sì	
SDK per Go V2 (1.x)	Sì	
SDK per Go 1.x (V1)	Sì	Per utilizzare le impostazioni dei `config` file condivisi, devi attivare il caricamento dal file di configurazione; vedi Sessioni.
SDK per Java 2.x	Sì	
SDK per Java 1.x	No	
SDK per 3.x JavaScript	Sì	
SDK per 2.x JavaScript	Sì	
SDK per Kotlin	Sì	
SDK per.NET 3.x	Sì	
SDK per PHP 3.x	Sì	
SDK per Python (Boto3)	Sì	

SDK	Su o	Note o ulteriori informazioni
SDK per Ruby 3.x	Sì	
SDK per Rust	Sì	
Strumenti per PowerShell	Sì	

Rilevamento di endpoint

Gli SDK utilizzano l'endpoint discovery per accedere agli endpoint dei servizi (URL per accedere a varie risorse), pur mantenendo la flessibilità necessaria per modificare gli URL AWS secondo necessità. In questo modo, il codice può rilevare automaticamente nuovi endpoint. Non esistono endpoint fissi per alcuni servizi. Al contrario, è possibile ottenere gli endpoint disponibili durante il runtime effettuando prima una richiesta per ottenere gli endpoint. Dopo aver recuperato gli endpoint disponibili, il codice utilizza quindi l'endpoint per accedere ad altre operazioni. Ad esempio, per Amazon Timestream, l'SDK effettua `DescribeEndpoints` una richiesta per recuperare gli endpoint disponibili e quindi utilizza tali endpoint per completare operazioni specifiche come o. `CreateDatabase` `CreateTable`

Configura questa funzionalità utilizzando quanto segue:

`endpoint_discovery_enabled`- impostazione dei AWS **`config`** file condivisi, **`AWS_ENABLE_ENDPOINT_DISCOVERY`**- variabile d'ambiente, **`aws.endpointDiscoveryEnabled`**- Proprietà del sistema JVM: solo Java/Kotlin, Per configurare il valore direttamente nel codice, consulta direttamente il tuo SDK specifico.

Attiva o disattiva il rilevamento degli endpoint per DynamoDB.

L'individuazione degli endpoint è richiesta in Timestream e facoltativa in Amazon DynamoDB. L'impostazione predefinita di questa impostazione è una delle due `true` o `false` dipende dal fatto che il servizio richieda il rilevamento degli endpoint. Le richieste Timestream sono predefinite su `true` e le richieste Amazon DynamoDB sono predefinite su. `false`

Valori validi:

- **`true`**— L'SDK dovrebbe tentare automaticamente di rilevare un endpoint per servizi in cui l'individuazione degli endpoint è facoltativa.

- **`false`**— L'SDK non dovrebbe tentare automaticamente di rilevare un endpoint per servizi in cui l'individuazione degli endpoint è facoltativa.

Compatibilità con gli SDK AWS

I seguenti SDK supportano le funzionalità e le impostazioni descritte in questo argomento. Vengono annotate eventuali eccezioni parziali. Tutte le impostazioni delle proprietà del sistema JVM sono supportate solo da AWS SDK for Java and the. SDK AWS for Kotlin

SDK	Su o	Note o ulteriori informazioni
AWS CLI v2	Sì	
SDK per C++	Sì	
SDK per Go V2 (1.x)	Sì	
SDK per Go 1.x (V1)	Sì	Per utilizzare le impostazioni dei `config` file condivisi, devi attivare il caricamento dal file di configurazione; vedi Sessioni.
SDK per Java 2.x	Sì	L'SDK for Java 2.x lo `AWS_ENDPOINT_DISCOVERY_ENABLED` utilizza per il nome della variabile di ambiente.
SDK per Java 1.x	Parzia	Proprietà di sistema JVM non supportata.
SDK per 3.x JavaScript	Sì	
SDK per 2.x JavaScript	Sì	
SDK per Kotlin	Sì	
SDK per.NET 3.x	Sì	
SDK per PHP 3.x	Sì	
SDK per Python (Boto3)	Sì	
SDK per Ruby 3.x	Sì	

SDK	Su o	Note o ulteriori informazioni
SDK per Rust	Parzia	Supportato solo per Timestream.
Strumenti per PowerShell	Sì	

Impostazioni generali di configurazione

Gli SDK supportano alcune impostazioni generali che configurano i comportamenti complessivi dell'SDK.

Configura questa funzionalità utilizzando quanto segue:

`api_versions`- impostazione dei AWS **`config`** file condivisi

Alcuni AWS servizi mantengono più versioni dell'API per supportare la compatibilità con le versioni precedenti. Per impostazione predefinita, SDK e AWS CLI operazioni utilizzano l'ultima versione API disponibile. Per richiedere una versione API specifica da utilizzare per le tue richieste, includi l'`api_versions`impostazione nel tuo profilo.

Valore predefinito: Nessuno. (L'ultima versione dell'API viene utilizzata dall'SDK.)

Valori validi: si tratta di un'impostazione annidata seguita da una o più righe rientrate, ciascuna delle quali identifica un AWS servizio e la versione dell'API da utilizzare. Consulta la documentazione del AWS servizio per capire quali versioni dell'API sono disponibili.

L'esempio imposta una versione API specifica per due AWS servizi nel `config` file. Queste versioni API vengono utilizzate solo per i comandi eseguiti nel profilo che contiene queste impostazioni. I comandi per qualsiasi altro servizio utilizzano la versione più recente dell'API di quel servizio.

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

`ca_bundle`- impostazione di AWS **`config`** file condivisi, **`AWS_CA_BUNDLE`**- variabile d'ambiente

Speciifica il percorso di un pacchetto di certificati personalizzato (un file con `.pem` estensione) da utilizzare per stabilire connessioni SSL/TLS.

Valore predefinito: nessuno

Valori validi: specificate il percorso completo o un nome di file di base. Se è presente un nome di file di base, il sistema tenta di trovare il programma all'interno delle cartelle specificate dalla variabile di PATH ambiente.

Esempio di impostazione di questo valore nel `config` file:

```
[default]
ca_bundle = dev/apps/ca-certs/cabundle-2019mar05.pem
```

Esempio in Linux/macOS di impostazione delle variabili di ambiente tramite riga di comando:

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

Esempio in Windows di impostazione delle variabili di ambiente tramite riga di comando:

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

`output`- impostazione di AWS **`config`** file condivisi

Specifica il modo in cui i risultati vengono formattati negli AWS SDK AWS CLI e negli altri strumenti.

Valore predefinito: `json`

Valori validi:

- **`json`**— L'output è formattato come stringa JSON.
- **`yaml`**— L'output è formattato come stringa YAML.
- **`yaml-stream`**— L'output viene trasmesso in streaming e formattato come stringa YAML. Lo streaming consente una gestione più rapida di tipi di dati di grandi dimensioni.
- **`text`**— L'output è formattato come più righe di valori di stringa separati da tabulazioni. Questo può essere utile per passare l'output a un elaboratore di testi, ad esempio`grep`, `sed` o. `awk`
- **`table`**— L'output viene formattato come tabella utilizzando i caratteri +|- per formare i bordi delle celle. In genere presenta le informazioni in un formato comprensibile molto più semplice da leggere rispetto ad altri, ma non altrettanto utile a livello programmatico.

`parameter_validation`- impostazione di file condivisi AWS **`config`**

Speciifica se l'SDK o lo strumento tenta di convalidare i parametri della riga di comando prima di inviarli all'endpoint del AWS servizio.

Valore predefinito: `true`

Valori validi:

- **`true`** – Il valore predefinito. L'SDK o lo strumento esegue la convalida lato client dei parametri della riga di comando. Ciò consente all'SDK o allo strumento di confermare la validità dei parametri e rileva alcuni errori. L'SDK o lo strumento possono rifiutare le richieste non valide prima di inviarle all'endpoint del servizio. AWS
- **`false`**— L'SDK o lo strumento non convalidano i parametri della riga di comando prima di inviarli all'endpoint del servizio. AWS L'endpoint del AWS servizio è responsabile della convalida di tutte le richieste e del rifiuto delle richieste non valide.

Compatibilità con gli SDK AWS

I seguenti SDK supportano le funzionalità e le impostazioni descritte in questo argomento. Vengono annotate eventuali eccezioni parziali. Tutte le impostazioni delle proprietà del sistema JVM sono supportate solo da AWS SDK for Java and the. SDK AWS for Kotlin

SDK	Su o	Note o ulteriori informazioni
AWS CLI v2	Parzia	`api_versions` non supportato.
SDK per C++	Sì	
SDK per Go V2 (1.x)	Parzia	`api_versions` e `parameter_validation` non supportato.
SDK per Go 1.x (V1)	Parzia	`api_versions` e `parameter_validation` non supportato. Per utilizzare le impostazioni dei `config` file condivisi, devi attivare il caricamento dal file di configurazione; vedi Sessioni.
SDK per Java 2.x	No	

SDK	Su o	Note o ulteriori informazioni
SDK per Java 1.x	No	
SDK per 3.x JavaScript	Sì	
SDK per 2.x JavaScript	Sì	
SDK per Kotlin	No	
SDK per.NET 3.x	No	
SDK per PHP 3.x	Sì	
SDK per Python (Boto3)	Sì	
SDK per Ruby 3.x	Sì	
SDK per Rust	No	
Strumenti per PowerShell	No	

Cliente IMDS

Gli SDK implementano un client Instance Metadata Service Version 2 (IMDSv2) utilizzando richieste orientate alla sessione. Per ulteriori informazioni su IMDSv2, consulta Use IMDSv2 nella Amazon EC2 User Guide o Use IMDSv2 nella Amazon EC2 User Guide. Il client IMDS è configurabile tramite un oggetto di configurazione del client disponibile nella base di codice SDK.

Configura questa funzionalità utilizzando quanto segue:

`retries`- membro dell'oggetto di configurazione del client

Il numero di tentativi di riprova aggiuntivi per ogni richiesta non riuscita.

Valore predefinito: 3

Valori validi: numero maggiore di 0.

`port`- membro dell'oggetto di configurazione del client

La porta per l'endpoint.

Valore predefinito: 80

Valori validi: Numero.

`token_ttl`- membro dell'oggetto di configurazione del client

Il TTL del token.

Valore predefinito: 21.600 secondi (6 ore, il tempo massimo assegnato).

Valori validi: Numero.

`endpoint`- membro dell'oggetto di configurazione del client

L'endpoint di IMDS.

Valore predefinito: se è `endpoint_mode` ugualeIPv4, l'endpoint predefinito è. `http://169.254.169.254` Se è `endpoint_mode` uguale, l'endpoint predefinito è. `IPv6` `http://[fd00:ec2::254]`

Valori validi: URI valido.

Le seguenti opzioni sono supportate dalla maggior parte degli SDK. Consulta la tua base di codice SDK specifica per i dettagli.

`endpoint_mode`- membro dell'oggetto di configurazione del client

La modalità endpoint di IMDS.

Valore predefinito: `IPv4`

Valori validi: `IPv4`, `IPv6`

`http_open_timeout`- membro dell'oggetto di configurazione del client (il nome può variare)

Il numero di secondi di attesa per l'apertura della connessione.

Valore predefinito: 1 secondo.

Valori validi: numero maggiore di 0.

`http_read_timeout`- membro dell'oggetto di configurazione del client (il nome può variare)

Il numero di secondi per la lettura di un blocco di dati.

Valore predefinito: 1 secondo.

Valori validi: numero maggiore di 0.

`http_debug_output`- membro dell'oggetto di configurazione del client (il nome può variare)

Imposta un flusso di output per il debug.

Valore predefinito: Nessuno.

Valori validi: un flusso di I/O valido, come STDOUT.

`backoff`- membro dell'oggetto di configurazione del client (il nome può variare)

Il numero di secondi trascorsi a dormire tra un tentativo e l'altro o la funzione di backoff fornita dal cliente per effettuare una chiamata. Ciò ha la precedenza sulla strategia di backoff esponenziale predefinita.

Valore predefinito: varia in base all'SDK.

Valori validi: varia in base all'SDK. Può essere un valore numerico o una chiamata a una funzione personalizzata.

Compatibilità con AWS gli SDK

I seguenti SDK supportano le funzionalità e le impostazioni descritte in questo argomento. Vengono annotate eventuali eccezioni parziali. Tutte le impostazioni delle proprietà del sistema JVM sono supportate solo da AWS SDK for Java and the. SDK AWS for Kotlin

SDK	Supporto	Note o ulteriori informazioni
AWS CLI v2	Sì	
SDK per C++	No	IMDSv2 utilizzato solo internamente. Per informazioni, consulta Provider di credenziali IMDS.
SDK per Go V2 (1.x)	Sì	

SDK	Su o	Note o ulteriori informazioni
SDK per Go 1.x (V1)	Sì	
SDK per Java 2.x	Sì	
SDK per Java 1.x	Sì	
SDK per 3.x JavaScript	Sì	
SDK per 2.x JavaScript	Sì	
SDK per Kotlin	Sì	
SDK per.NET 3.x	Sì	
SDK per PHP 3.x	Sì	
SDK per Python (Boto3)	Sì	
SDK per Ruby 3.x	Sì	
SDK per Rust	Sì	
Strumenti per PowerShell	Sì	

Comportamento di ripetizione

Il comportamento Riprova include le impostazioni relative al modo in cui gli SDK tentano di ripristinare gli errori derivanti dalle richieste effettuate a. Servizi AWS

Configura questa funzionalità utilizzando quanto segue:

`max_attempts`- impostazione dei AWS **`config`** file condivisi, **`AWS_MAX_ATTEMPTS`**- variabile d'ambiente, **`aws.maxAttempts`**- Proprietà del sistema JVM: solo Java/Kotlin

Speciifica il numero massimo di tentativi da effettuare su una richiesta.

Valore predefinito: se questo valore non è specificato, il valore predefinito dipende dal valore dell'`retry_mode`impostazione:

- Se lo `retry_mode` è`legacy`: utilizza un valore predefinito specifico per il tuo SDK (consulta la guida SDK specifica o la base di codice dell'SDK per `max_attempts` i valori predefiniti).
- Se lo `retry_mode` è`standard`: effettua tre tentativi.
- Se lo `retry_mode` è`adaptive`: effettua tre tentativi.

Valori validi: numero maggiore di 0.

`retry_mode`- impostazione condivisa AWS **`config`** dei file, **`AWS_RETRY_MODE`**- variabile d'ambiente, **`aws.retryMode`**- Proprietà del sistema JVM: solo Java/Kotlin

Specifica in che modo l'SDK o lo strumento di sviluppo tenta di riprovare.

Valore predefinito: `legacy` è la strategia di ripetizione dei tentativi predefinita.

Valori validi:

- `legacy`— Specifico per il tuo SDK (consulta la tua guida SDK specifica o la base di codice dell'SDK).
- `standard`— Il set standard di regole di riprova tra gli SDK. AWS Questa modalità include un set standard di errori che vengono ripetuti e il supporto per le quote di tentativi. Il numero massimo predefinito di tentativi con questa modalità è tre, a meno che non `max_attempts` sia configurata in modo esplicito.
- `adaptive`— Una modalità sperimentale di ripetizione che include le funzionalità della modalità standard ma include la limitazione automatica sul lato client. Poiché questa modalità è sperimentale, potrebbe modificare il comportamento in futuro.

Scelta tra le **`standard`** modalità e **`adaptive`** riprova

Ti consigliamo di utilizzare la modalità `standard` Riprova a meno che tu non sia sicuro che il tuo utilizzo sia più adatto. `adaptive`

> **Note**
>
> La `adaptive` modalità presuppone che stiate raggruppando i client in base all'ambito in cui il servizio di backend può limitare le richieste. Se non lo fai, le limitazioni di una risorsa potrebbero ritardare le richieste di una risorsa non correlata se utilizzi lo stesso client per entrambe le risorse.

Standard	Adattabile
Casi d'uso delle applicazioni: tutti.	Casi d'uso dell'applicazione: 1. Non sensibile alla latenza. 2. Il client accede solo a una singola risorsa oppure state fornendo la logica per raggruppare i client separatamente in base alla risorsa di servizio a cui si accede.
Supporta l'interruzione del circuito per impedire che l'SDK riprovi durante le interruzioni.	Supporta l'interruzione del circuito per impedire all'SDK di riprovare durante le interruzioni.
Utilizza un backoff esponenziale con jitterato in caso di guasti.	Utilizza durate di backoff dinamiche per cercare di ridurre al minimo il numero di richieste non riuscite, in cambio del potenziale aumento della latenza.
Non ritarda mai il primo tentativo di richiesta, ma solo i tentativi successivi.	Può rallentare o ritardare il tentativo di richiesta iniziale.

Se si sceglie di utilizzare la `adaptive` modalità, l'applicazione deve creare client progettati in base a ciascuna risorsa che potrebbe essere limitata. Una risorsa, in questo caso, è ottimizzata in modo più preciso e non si limita a pensare a ciascuna di esse. Servizio AWS Servizi AWS possono avere dimensioni aggiuntive che utilizzano per limitare le richieste. Usiamo il servizio Amazon DynamoDB come esempio. DynamoDB Regione AWS utilizza inoltre la tabella a cui si accede per limitare le richieste. Ciò significa che una tabella a cui accede il codice potrebbe essere limitata più di altre. Se il codice utilizza lo stesso client per accedere a tutte le tabelle e le richieste a una di tali tabelle sono limitate, la modalità di riprova adattiva ridurrà la frequenza di richieste per tutte le tabelle. Il tuo codice dovrebbe essere progettato per avere un client per egion-and-table coppia R. Se riscontri una latenza inaspettata durante l'utilizzo della `adaptive` modalità, consulta la guida alla AWS documentazione specifica per il servizio che stai utilizzando.

Dettagli sull'implementazione della modalità Riprova

Di seguito è riportato lo pseudocodice di alto livello per entrambe le modalità e retry: `standard` `adaptive`

```
MakeSDKRequest() {
  attempts = 0
  loop {
    GetSendToken()
    response = SendHTTPRequest()
    RequestBookkeeping(response)
    if not Retryable(response)
      return response
    attempts += 1
    if attempts >= MAX_ATTEMPTS:
      return response
    if not HasRetryQuota(response)
      return response
    delay = ExponentialBackoff(attempts)
    sleep(delay)
  }
}
```

Di seguito sono riportati ulteriori dettagli sui componenti utilizzati nello pseudocodice:

`GetSendToken`:

I token bucket vengono utilizzati solo in modalità riprova. `adaptive` I token bucket impongono una frequenza massima di richieste richiedendo la disponibilità di un token per avviare una richiesta. Il client SDK è configurabile per fallire rapidamente la richiesta o bloccarla finché non diventa disponibile un token.

Client Side Rate Limiting è un algoritmo che inizialmente consente di effettuare richieste in qualsiasi momento, fino al limite consentito dal token. Tuttavia, dopo che viene rilevata una risposta limitata, il client rate-of-request viene limitato di conseguenza. La soglia di token viene inoltre aumentata di conseguenza se vengono ricevute risposte positive.

Grazie alla limitazione adattiva della velocità, gli SDK possono rallentare la velocità di invio delle richieste per soddisfare meglio la capacità di. Servizi AWS

`SendHTTPRequest`:

La maggior parte degli AWS SDK utilizza una libreria HTTP che utilizza pool di connessioni in modo da poter riutilizzare una connessione esistente quando si effettua una richiesta HTTP. In genere, le connessioni vengono riutilizzate quando si riprovano le richieste a causa di errori di limitazione. Le richieste non vengono riutilizzate quando si riprova a causa di errori temporanei.

`RequestBookkeeping`:

La quota di tentativi deve essere aggiornata se la richiesta ha esito positivo. Solo per la modalità `adaptive` riprova, la variabile di stato `maxsendrate` viene aggiornata in base al tipo di risposta ricevuta.

`Retryable`:

Questo passaggio determina se una risposta può essere ritentata in base a quanto segue:

- Codice di stato HTTP .
- Il codice di errore restituito dal servizio.
- Errori di connessione, definiti come qualsiasi errore ricevuto dall'SDK in cui non viene ricevuta una risposta HTTP dal servizio.

Gli errori transitori (codici di stato HTTP 400, 408, 500, 502, 503 e 504) e gli errori di limitazione (codici di stato HTTP 400, 403, 429, 502, 503 e 509) possono essere tutti potenzialmente ritentati. Il comportamento dei nuovi tentativi dell'SDK viene determinato in combinazione con codici di errore o altri dati del servizio.

`MAX_ATTEMPTS`:

Specificato dall'impostazione del `config` file o dalla variabile di ambiente.

`HasRetryQuota`

Questo passaggio limita le richieste di nuovi tentativi richiedendo che un token sia disponibile nel bucket delle quote di tentativi. I gruppi di quote per nuovi tentativi sono un meccanismo per impedire nuovi tentativi che difficilmente avranno successo. Queste quote dipendono dall'SDK, spesso dipendono dal client e talvolta dipendono anche dagli endpoint del servizio. I token di quota di tentativi disponibili vengono rimossi quando le richieste hanno esito negativo per vari motivi e ripristinati quando hanno esito positivo. Quando non rimangono token, il ciclo di ripetizione dei tentativi viene chiuso.

`ExponentialBackoff`

Per un errore che può essere riprovato, il ritardo tra i tentativi viene calcolato utilizzando un backoff esponenziale troncato. Gli SDK utilizzano un backoff esponenziale binario troncato con jitter. L'algoritmo seguente mostra come viene definita la quantità di tempo per dormire, in secondi, per una risposta a una richiesta: `i`

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

Nell'algoritmo precedente, si applicano i seguenti valori:

`b = random number within the range of: 0 <= b <= 1`

`r = 2`

`MAX_BACKOFF = 20 seconds`per la maggior parte degli SDK. Per conferma, consulta la guida SDK o il codice sorgente specifici.

Compatibilità con AWS gli SDK

I seguenti SDK supportano le funzionalità e le impostazioni descritte in questo argomento. Vengono annotate eventuali eccezioni parziali. Tutte le impostazioni delle proprietà del sistema JVM sono supportate solo da AWS SDK for Java and the. SDK AWS for Kotlin

SDK	Supporto	Note o ulteriori informazioni
AWS CLI v2	Sì	
SDK per C++	Sì	
SDK per Go V2 (1.x)	Sì	
SDK per Go 1.x (V1)	No	
SDK per Java 2.x	Sì	
SDK per Java 1.x	Sì	Proprietà del sistema JVM: usa `com.amazonaws.sdk.maxAttempts` invece di`aws.maxAttempts` ; usa invece di. `com.amazonaws.sdk.retryMode` `aws.retryMode`
SDK per 3.x JavaScript	Sì	
SDK per 2.x JavaScript	No	Supporta un numero massimo di tentativi, un backoff esponenziale con jitter e un'opzione per un metodo personalizzato per il backoff dei tentativi.

SDK	Su o	Note o ulteriori informazioni
SDK per Kotlin	Sì	
SDK per.NET 3.x	Sì	
SDK per PHP 3.x	Sì	
SDK per Python (Boto3)	Sì	
SDK per Ruby 3.x	Sì	
SDK per Rust	Sì	
Strumenti per PowerShell	Sì	

Richiesta di compressione

AWS Gli SDK e gli strumenti possono comprimere automaticamente i payload quando si inviano richieste al supporto che riceve payload compressi. Servizi AWS La compressione del payload sul client prima di inviarlo a un servizio può ridurre il numero complessivo di richieste e la larghezza di banda necessari per inviare dati al servizio, nonché ridurre le richieste non riuscite a causa delle limitazioni del servizio sulla dimensione del payload. Per la compressione, l'SDK o lo strumento seleziona un algoritmo di codifica supportato sia dal servizio che dall'SDK. Tuttavia, l'elenco attuale delle possibili codifiche è costituito solo da gzip, ma potrebbe espandersi in futuro.

La compressione delle richieste può essere particolarmente utile se l'applicazione utilizza Amazon CloudWatch. CloudWatch è un servizio di monitoraggio e osservabilità che raccoglie dati operativi e di monitoraggio sotto forma di log, metriche ed eventi. Un esempio di funzionamento di servizio che supporta la compressione è CloudWatch il metodo API. PutMetricData

Configura questa funzionalità utilizzando quanto segue:

`disable_request_compression`- impostazione dei AWS **`config`** file condivisi, **`AWS_DISABLE_REQUEST_COMPRESSION`**- variabile d'ambiente, **`aws.disableRequestCompression`**- Proprietà del sistema JVM: solo Java/Kotlin

Attiva o disattiva se l'SDK o lo strumento comprimeranno un payload prima di inviare una richiesta.

Valore predefinito: `false`

Valori validi:

- **`true`**— Disattiva la compressione delle richieste.
- **`false`**— Usa la compressione delle richieste quando possibile.

`request_min_compression_size_bytes`- impostazione dei AWS **`config`** file condivisi, **`AWS_REQUEST_MIN_COMPRESSION_SIZE_BYTES`**- variabile d'ambiente, **`aws.requestMinCompressionSizeBytes`**- Proprietà del sistema JVM: solo Java/Kotlin

Imposta la dimensione minima in byte del corpo della richiesta che l'SDK o lo strumento devono comprimere. I carichi utili di piccole dimensioni possono allungarsi quando vengono compressi, quindi esiste un limite inferiore in base al quale è opportuno eseguire la compressione. Questo valore è inclusivo, viene compressa una dimensione della richiesta maggiore o uguale al valore.

Valore predefinito: 10240 byte

Valori validi: valore intero compreso tra 0 e 10485760 byte inclusi.

AWS Compatibilità con gli SDK

I seguenti SDK supportano le funzionalità e le impostazioni descritte in questo argomento. Vengono annotate eventuali eccezioni parziali. Tutte le impostazioni delle proprietà del sistema JVM sono supportate solo da AWS SDK for Java and the. SDK AWS for Kotlin

SDK	Supporto	Note o ulteriori informazioni
AWS CLI v2	Sì	
SDK per C++	Sì	

SDK	Su o	Note o ulteriori informazioni
SDK per Go V2 (1.x)	Sì	
SDK per Go 1.x (V1)	No	
SDK per Java 2.x	Sì	
SDK per Java 1.x	No	
SDK per 3.x JavaScript	Sì	
SDK per 2.x JavaScript	No	
SDK per Kotlin	Sì	
SDK per.NET 3.x	Sì	
SDK per PHP 3.x	Sì	
SDK per Python (Boto3)	Sì	
SDK per Ruby 3.x	Sì	
SDK per Rust	Sì	
Strumenti per PowerShell	Sì	

Endpoint specifici del servizio

La configurazione degli endpoint specifica per il servizio offre la possibilità di utilizzare un endpoint di propria scelta per le richieste API e di mantenere tale scelta. Queste impostazioni offrono la flessibilità necessaria per supportare endpoint locali, endpoint VPC e ambienti di sviluppo locale di terze parti AWS . È possibile utilizzare endpoint diversi per ambienti di test e produzione. È possibile specificare un URL di endpoint per singoli utenti. Servizi AWS

Configura questa funzionalità utilizzando quanto segue:

`endpoint_url`- impostazione dei AWS **`config`** file condivisi, **`AWS_ENDPOINT_URL`**- variabile d'ambiente, **`aws.endpointUrl`**- Proprietà del sistema JVM: solo Java/Kotlin

Se specificata direttamente all'interno di un profilo o come variabile di ambiente, questa impostazione specifica l'endpoint utilizzato per tutte le richieste di servizio. Questo endpoint viene sovrascritto da qualsiasi endpoint configurato specifico del servizio.

È inoltre possibile utilizzare questa impostazione all'interno di una `services` sezione di un AWS `config` file condiviso per impostare un endpoint personalizzato per un servizio specifico. Per un elenco di tutte le chiavi identificative del servizio da utilizzare per le sottosezioni all'interno della `services` sezione, vedere. Identificatori per endpoint specifici del servizio

Valore predefinito: `none`

Valori validi: un URL che include lo schema e l'host per l'endpoint. L'URL può facoltativamente contenere un componente del percorso che contiene uno o più segmenti di percorso.

`AWS_ENDPOINT_URL_<SERVICE>`- variabile di ambiente, **`aws.endpointUrl<ServiceName>`**- Proprietà del sistema JVM: solo Java/Kotlin

`AWS_ENDPOINT_URL_<SERVICE>`, `<SERVICE>` dov'è l' Servizio AWS identificatore, imposta un endpoint personalizzato per un servizio specifico. Per un elenco di tutte le variabili di ambiente specifiche del servizio, vedere. Identificatori per endpoint specifici del servizio

Questo endpoint specifico del servizio sostituisce qualsiasi endpoint globale impostato. `AWS_ENDPOINT_URL`

Valore predefinito: `none`

Valori validi: un URL che include lo schema e l'host per l'endpoint. L'URL può facoltativamente contenere un componente del percorso che contiene uno o più segmenti di percorso.

`ignore_configured_endpoint_urls`- impostazione condivisa AWS **`config`** dei file, **`AWS_IGNORE_CONFIGURED_ENDPOINT_URLS`**- variabile d'ambiente, **`aws.ignoreConfiguredEndpointUrls`**- Proprietà del sistema JVM: solo Java/Kotlin

Questa impostazione viene utilizzata per ignorare tutte le configurazioni personalizzate degli endpoint.

Tieni presente che qualsiasi endpoint esplicito impostato nel codice o su uno stesso client di servizio viene utilizzato indipendentemente da questa impostazione. Ad esempio, l'inclusione del

parametro della riga di `--endpoint-url` comando con un AWS CLI comando o il passaggio di un URL di endpoint a un costruttore del client avrà sempre effetto.

Valore predefinito: `false`

Valori validi:

- **`true`**— L'SDK o lo strumento non leggono alcuna opzione di configurazione personalizzata dal `config` file condiviso o dalle variabili di ambiente per l'impostazione di un URL dell'endpoint.
- **`false`**— L'SDK o lo strumento utilizza tutti gli endpoint disponibili forniti dall'utente dal `config` file condiviso o dalle variabili di ambiente.

Configura gli endpoint utilizzando variabili di ambiente

Per indirizzare le richieste di tutti i servizi a un URL endpoint personalizzato, imposta la variabile di ambiente `AWS_ENDPOINT_URL` globale.

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

Per indirizzare le richieste per un URL di endpoint specifico Servizio AWS a un URL di endpoint personalizzato, utilizza la variabile di `AWS_ENDPOINT_URL_<SERVICE>` ambiente. Amazon DynamoDB ha un`serviceId`. `DynamoDB` Per questo servizio, la variabile di ambiente dell'URL dell'endpoint è`AWS_ENDPOINT_URL_DYNAMODB`. Questo endpoint ha la precedenza sull'endpoint globale impostato per questo servizio. `AWS_ENDPOINT_URL`

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

Come altro esempio, AWS Elastic Beanstalk ha un di. `serviceId` `Elastic Beanstalk` L' Servizio AWS identificatore si basa sul modello API `serviceId` sostituendo tutti gli spazi con caratteri di sottolineatura e tutte le lettere maiuscole. Per impostare l'endpoint per questo servizio, la variabile di ambiente corrispondente è. `AWS_ENDPOINT_URL_ELASTIC_BEANSTALK` Per un elenco di tutte le variabili di ambiente specifiche del servizio, vedere. Identificatori per endpoint specifici del servizio

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

Configurare gli endpoint utilizzando il file condiviso **`config`**

Nel `config` file condiviso, `endpoint_url` viene utilizzato in luoghi diversi per funzionalità diverse.

- `endpoint_url`specificato direttamente all'interno di a `profile` rende quell'endpoint l'endpoint globale.
- `endpoint_url`annidato sotto una chiave identificativa del servizio all'interno di una `services` sezione fa sì che l'endpoint si applichi alle richieste fatte solo a quel servizio. Per i dettagli sulla definizione di una `services` sezione nel `config` file condiviso, consulta. Formato del file di configurazione

L'esempio seguente utilizza una `services` definizione per configurare un URL di endpoint specifico del servizio da utilizzare per Amazon S3 e un endpoint globale personalizzato da utilizzare per tutti gli altri servizi:

```
[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
```

Un singolo profilo può configurare gli endpoint per più servizi. Questo esempio mostra come impostare gli URL degli endpoint specifici del servizio per Amazon S3 e nello stesso profilo. AWS Elastic Beanstalk AWS Elastic Beanstalk ha un. `serviceId` `Elastic Beanstalk` L' Servizio AWS identificatore si basa sul modello API `serviceId` sostituendo tutti gli spazi con caratteri di sottolineatura e tutte le lettere minuscole. Pertanto, la chiave identificativa del servizio diventa `elastic_beanstalk` e le impostazioni per questo servizio iniziano sulla linea. `elastic_beanstalk =` Per un elenco di tutte le chiavi identificative del servizio da utilizzare nella `services` sezione, vedere. Identificatori per endpoint specifici del servizio

```
[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
```

La sezione di configurazione del servizio può essere utilizzata da più profili. Ad esempio, due profili possono utilizzare la stessa `services` definizione modificando altre proprietà del profilo:

```
[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
```

Configura gli endpoint nei profili utilizzando credenziali basate sui ruoli

Se il tuo profilo ha credenziali basate sui ruoli configurate tramite un `source_profile` parametro per IAM Assume Role Functionality, l'SDK utilizza solo le configurazioni di servizio per il profilo specificato. Non utilizza profili concatenati a ruoli. Ad esempio, utilizzando il seguente `config` file condiviso:

```
[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
```

Se usi il profilo `B` ed effettui una chiamata nel codice verso Amazon EC2, l'endpoint si risolve come. `https://profile-b-ec2-endpoint.aws` Se il codice invia una richiesta a qualsiasi altro servizio, la risoluzione dell'endpoint non seguirà alcuna logica personalizzata. L'endpoint non si risolve nell'endpoint globale definito nel profilo. `A` Affinché un endpoint globale abbia effetto sul profilo`B`, è necessario `endpoint_url` impostarlo direttamente all'interno del profilo. `B` Per ulteriori informazioni in merito all'impostazione `source_profile`, consulta Assumi il ruolo di fornitore di credenziali.

Precedenza delle impostazioni

Le impostazioni di questa funzionalità possono essere utilizzate contemporaneamente, ma solo un valore avrà la priorità per servizio. Per le chiamate API effettuate verso un determinato valore Servizio AWS, viene utilizzato il seguente ordine per selezionare un valore:

1. Qualsiasi impostazione esplicita impostata nel codice o su un client di servizio stesso ha la precedenza su qualsiasi altra cosa.
   - Per il AWS CLI, questo è il valore fornito dal parametro della `--endpoint-url` riga di comando. Per un SDK, le assegnazioni esplicite possono assumere la forma di un parametro impostato quando si crea un'istanza di un client o di un Servizio AWS oggetto di configurazione.
2. Il valore fornito da una variabile di ambiente specifica del servizio, ad esempio. `AWS_ENDPOINT_URL_DYNAMODB`
3. Il valore fornito dalla variabile di ambiente `AWS_ENDPOINT_URL` globale dell'endpoint.
4. Il valore fornito dall'`endpoint_url`impostazione annidata in una chiave di identificazione del servizio all'interno di una `services` sezione del file condiviso. `config`
5. Il valore fornito dall'`endpoint_url`impostazione specificato direttamente all'interno `profile` di uno dei file condivisi`config`.
6. Qualsiasi URL di endpoint predefinito per il rispettivo Servizio AWS viene utilizzato per ultimo.

Compatibilità con AWS gli SDK

I seguenti SDK supportano le funzionalità e le impostazioni descritte in questo argomento. Vengono annotate eventuali eccezioni parziali. Tutte le impostazioni delle proprietà del sistema JVM sono supportate solo da AWS SDK for Java and the. SDK AWS for Kotlin

SDK	Su o	Note o ulteriori informazioni
AWS CLI v2	Sì	
SDK per C++	No	
SDK per Go V2 (1.x)	Sì	
SDK per Go 1.x (V1)	No	

SDK	Su o	Note o ulteriori informazioni
SDK per Java 2.x	No	
SDK per Java 1.x	No	
SDK per 3.x JavaScript	Sì	
SDK per 2.x JavaScript	No	
SDK per Kotlin	Sì	
SDK per.NET 3.x	Sì	
SDK per PHP 3.x	Sì	
SDK per Python (Boto3)	Sì	
SDK per Ruby 3.x	Sì	
SDK per Rust	No	
Strumenti per PowerShell	Sì	

Identificatori per endpoint specifici del servizio

Per informazioni su come e dove utilizzare gli identificatori nella tabella seguente, vedere. Endpoint specifici del servizio

serviceId	Cl id at de se pe il fil cc A\ **co**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
AccessAnalyzer	ac ly	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	a: _I	AWS_ENDPOINT_URL_ALEXA_FOR_BUSINESS
amp	ar	AWS_ENDPOINT_URL_AMP
Amplify	ar	AWS_ENDPOINT_URL_AMPLIFY
AmplifyBackend	ar cl	AWS_ENDPOINT_URL_AMPLIFYBACKEND
AmplifyUIBuilder	ar bu	AWS_ENDPOINT_URL_AMPLIFYUIBUILDER
API Gateway	ap ay	AWS_ENDPOINT_URL_API_GATEWAY

serviceId	Cl id at de se pe il fil cc A\ **cc**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
ApiGatewayManagement ntApi	ap yr nt	AWS_ENDPOINT_URL_APIGATEWAYMANAGEMENTAPI
ApiGatewayV2	ap yv	AWS_ENDPOINT_URL_APIGATEWAYV2
AppConfig	ap	AWS_ENDPOINT_URL_APPCONFIG
AppConfigData	ap da	AWS_ENDPOINT_URL_APPCONFIGDATA
AppFabric	ap	AWS_ENDPOINT_URL_APPFABRIC
Appflow	ap	AWS_ENDPOINT_URL_APPFLOW
AppIntegrations	ap at	AWS_ENDPOINT_URL_APPINTEGRATIONS
Application Auto Scaling	ap or ca	AWS_ENDPOINT_URL_APPLICATION_AUTO_SCALING

serviceId	Cl id at de se pe il fil cc A\ **cc**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
Application Insights	aı or ts	AWS_ENDPOINT_URL_APPLICATION_INSIGHTS
ApplicationCostPro filer	aı or f:	AWS_ENDPOINT_URL_APPLICATIONCOSTPROFILER
App Mesh	aı	AWS_ENDPOINT_URL_APP_MESH
AppRunner	aı	AWS_ENDPOINT_URL_APPRUNNER
AppStream	aı	AWS_ENDPOINT_URL_APPSTREAM
AppSync	aı	AWS_ENDPOINT_URL_APPSYNC
ARC Zonal Shift	a: _s	AWS_ENDPOINT_URL_ARC_ZONAL_SHIFT
Artifact	a:	AWS_ENDPOINT_URL_ARTIFACT
Athena	at	AWS_ENDPOINT_URL_ATHENA
AuditManager	au ge	AWS_ENDPOINT_URL_AUDITMANAGER

serviceId	Cl id at de se pe il fil cc A\ **ce**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
Auto Scaling	au in	AWS_ENDPOINT_URL_AUTO_SCALING
Auto Scaling Plans	au in	AWS_ENDPOINT_URL_AUTO_SCALING_PLANS
b2bi	b2	AWS_ENDPOINT_URL_B2BI
Backup	ba	AWS_ENDPOINT_URL_BACKUP
Backup Gateway	ba te	AWS_ENDPOINT_URL_BACKUP_GATEWAY
BackupStorage	ba ra	AWS_ENDPOINT_URL_BACKUPSTORAGE
Batch	ba	AWS_ENDPOINT_URL_BATCH
BCM Data Exports	bc ex	AWS_ENDPOINT_URL_BCM_DATA_EXPORTS
Bedrock	be	AWS_ENDPOINT_URL_BEDROCK
Bedrock Agent	be ge	AWS_ENDPOINT_URL_BEDROCK_AGENT

serviceId	Cl id at de se pe il fil cc A\ **cc**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
Bedrock Agent Runtime	be ge ir	AWS_ENDPOINT_URL_BEDROCK_AGENT_RUNTIME
Bedrock Runtime	be ur	AWS_ENDPOINT_URL_BEDROCK_RUNTIME
billingconductor	bi nc	AWS_ENDPOINT_URL_BILLINGCONDUCTOR
Braket	bi	AWS_ENDPOINT_URL_BRAKET
Budgets	bu	AWS_ENDPOINT_URL_BUDGETS
Cost Explorer	co o:	AWS_ENDPOINT_URL_COST_EXPLORER
chatbot	ch	AWS_ENDPOINT_URL_CHATBOT
Chime	ch	AWS_ENDPOINT_URL_CHIME
Chime SDK Identity	ch _i	AWS_ENDPOINT_URL_CHIME_SDK_IDENTITY

serviceId	Cl id at de se pe il fil cc A\ **cc**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
Chime SDK Media Pipelines	ch _r pe	AWS_ENDPOINT_URL_CHIME_SDK_MEDIA_PIPELINES
Chime SDK Meetings	ch _r	AWS_ENDPOINT_URL_CHIME_SDK_MEETINGS
Chime SDK Messaging	ch _r g	AWS_ENDPOINT_URL_CHIME_SDK_MESSAGING
Chime SDK Voice	ch _\	AWS_ENDPOINT_URL_CHIME_SDK_VOICE
CleanRooms	c: s	AWS_ENDPOINT_URL_CLEANROOMS
CleanRoomsML	c: sr	AWS_ENDPOINT_URL_CLEANROOMSML
Cloud9	c:	AWS_ENDPOINT_URL_CLOUD9
CloudControl	c: rc	AWS_ENDPOINT_URL_CLOUDCONTROL

serviceId	Cl id at de se pe il fil cc A\ **cc**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
CloudDirectory	c: ct	AWS_ENDPOINT_URL_CLOUDDIRECTORY
CloudFormation	c: at	AWS_ENDPOINT_URL_CLOUDFORMATION
CloudFront	c: t	AWS_ENDPOINT_URL_CLOUDFRONT
CloudFront KeyValueStore	c: t_ es	AWS_ENDPOINT_URL_CLOUDFRONT_KEYVALUESTORE
CloudHSM	c:	AWS_ENDPOINT_URL_CLOUDHSM
CloudHSM V2	c: v2	AWS_ENDPOINT_URL_CLOUDHSM_V2
CloudSearch	c: ch	AWS_ENDPOINT_URL_CLOUDSEARCH
CloudSearch Domain	c: ch	AWS_ENDPOINT_URL_CLOUDSEARCH_DOMAIN

serviceId	Cl id at de se pe il fil co A\ **co**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
CloudTrail	cl l	AWS_ENDPOINT_URL_CLOUDTRAIL
CloudTrail Data	cl l_	AWS_ENDPOINT_URL_CLOUDTRAIL_DATA
CloudWatch	cl h	AWS_ENDPOINT_URL_CLOUDWATCH
codeartifact	co ac	AWS_ENDPOINT_URL_CODEARTIFACT
CodeBuild	co	AWS_ENDPOINT_URL_CODEBUILD
CodeCatalyst	co ys	AWS_ENDPOINT_URL_CODECATALYST
CodeCommit	co t	AWS_ENDPOINT_URL_CODECOMMIT
CodeDeploy	co y	AWS_ENDPOINT_URL_CODEDEPLOY
CodeGuru Reviewer	co re	AWS_ENDPOINT_URL_CODEGURU_REVIEWER

serviceId	Cl id at de se pe il fil co A\ **co**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
CodeGuru Security	co se	AWS_ENDPOINT_URL_CODEGURU_SECURITY
CodeGuruProfiler	co ro	AWS_ENDPOINT_URL_CODEGURUPROFILER
CodePipeline	co ir	AWS_ENDPOINT_URL_CODEPIPELINE
CodeStar	co	AWS_ENDPOINT_URL_CODESTAR
CodeStar connections	co co ns	AWS_ENDPOINT_URL_CODESTAR_CONNECTIONS
codestar notifications	co no ic	AWS_ENDPOINT_URL_CODESTAR_NOTIFICATIONS
Cognito Identity	co de	AWS_ENDPOINT_URL_COGNITO_IDENTITY
Cognito Identity Provider	co de ro	AWS_ENDPOINT_URL_COGNITO_IDENTITY_PROVIDER

serviceId	Cl id at de se pe il fil co AV **co**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
Cognito Sync	co yr	AWS_ENDPOINT_URL_COGNITO_SYNC
Comprehend	co d	AWS_ENDPOINT_URL_COMPREHEND
ComprehendMedical	co dr	AWS_ENDPOINT_URL_COMPREHENDMEDICAL
Compute Optimizer	co pt	AWS_ENDPOINT_URL_COMPUTE_OPTIMIZER
Config Service	co rv	AWS_ENDPOINT_URL_CONFIG_SERVICE
Connect	co	AWS_ENDPOINT_URL_CONNECT
Connect Contact Lens	co or ns	AWS_ENDPOINT_URL_CONNECT_CONTACT_LENS
ConnectCampaigns	co mp	AWS_ENDPOINT_URL_CONNECTCAMPAIGNS
ConnectCases	co se	AWS_ENDPOINT_URL_CONNECTCASES

serviceId	Cl id at de se pe il fil co AV **co**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
ConnectParticipant	co rt	AWS_ENDPOINT_URL_CONNECTPARTICIPANT
ControlTower	co we	AWS_ENDPOINT_URL_CONTROLTOWER
Cost Optimization Hub	co m: hu	AWS_ENDPOINT_URL_COST_OPTIMIZATION_HUB
Cost and Usage Report Service	co us o: ce	AWS_ENDPOINT_URL_COST_AND_USAGE_REPO RT_SERVICE
Customer Profiles	cu p:	AWS_ENDPOINT_URL_CUSTOMER_PROFILES
DataBrew	da	AWS_ENDPOINT_URL_DATABREW
DataExchange	da ng	AWS_ENDPOINT_URL_DATAEXCHANGE
Data Pipeline	da l:	AWS_ENDPOINT_URL_DATA_PIPELINE

serviceId	Cl id at de se pe il fil cc A\ **ce**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
DataSync	da	AWS_ENDPOINT_URL_DATASYNC
DataZone	da	AWS_ENDPOINT_URL_DATAZONE
DAX	da	AWS_ENDPOINT_URL_DAX
Detective	de	AWS_ENDPOINT_URL_DETECTIVE
Device Farm	de rr	AWS_ENDPOINT_URL_DEVICE_FARM
DevOps Guru	de ru	AWS_ENDPOINT_URL_DEVOPS_GURU
Direct Connect	di nr	AWS_ENDPOINT_URL_DIRECT_CONNECT
Application Discovery Service	ap or e: ce	AWS_ENDPOINT_URL_APPLICATION_DISCOVERY_SERVICE
DLM	dl	AWS_ENDPOINT_URL_DLM

serviceId	Cl id at de se pe il fil cc A\ **co**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
Database Migration Service	da m: _s	AWS_ENDPOINT_URL_DATABASE_MIGRATION_ SERVICE
DocDB	dc	AWS_ENDPOINT_URL_DOCDB
DocDB Elastic	dc st	AWS_ENDPOINT_URL_DOCDB_ELASTIC
drs	dr	AWS_ENDPOINT_URL_DRS
Directory Service	di _s	AWS_ENDPOINT_URL_DIRECTORY_SERVICE
DynamoDB	dy	AWS_ENDPOINT_URL_DYNAMODB
DynamoDB Streams	dy st	AWS_ENDPOINT_URL_DYNAMODB_STREAMS
EBS	eb	AWS_ENDPOINT_URL_EBS
EC2	ec	AWS_ENDPOINT_URL_EC2
EC2 Instance Connect	ec nc ct	AWS_ENDPOINT_URL_EC2_INSTANCE_CONNECT

serviceId	Cl id at de se pe il fil cc A\ **cc**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
ECR	ec	AWS_ENDPOINT_URL_ECR
ECR PUBLIC	ec c	AWS_ENDPOINT_URL_ECR_PUBLIC
ECS	ec	AWS_ENDPOINT_URL_ECS
EFS	ef	AWS_ENDPOINT_URL_EFS
EKS	el	AWS_ENDPOINT_URL_EKS
EKS Auth	el	AWS_ENDPOINT_URL_EKS_AUTH
Elastic Inference	el nf	AWS_ENDPOINT_URL_ELASTIC_INFERENCE
ElastiCache	el he	AWS_ENDPOINT_URL_ELASTICACHE
Elastic Beanstalk	el ea	AWS_ENDPOINT_URL_ELASTIC_BEANSTALK
Elastic Transcoder	el ra	AWS_ENDPOINT_URL_ELASTIC_TRANSCODER

serviceId	Cl id at de se pe il fil cc A\ **ce**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
Elastic Load Balancing	el oa c:	AWS_ENDPOINT_URL_ELASTIC_LOAD_BALANCING
Elastic Load Balancing v2	el oa c:	AWS_ENDPOINT_URL_ELASTIC_LOAD_BALANCING_V2
EMR	er	AWS_ENDPOINT_URL_EMR
EMR containers	er ir	AWS_ENDPOINT_URL_EMR_CONTAINERS
EMR Serverless	er r:	AWS_ENDPOINT_URL_EMR_SERVERLESS
EntityResolution	er o:	AWS_ENDPOINT_URL_ENTITYRESOLUTION
Elasticsearch Service	el a: ic	AWS_ENDPOINT_URL_ELASTICSEARCH_SERVICE
EventBridge	ev ge	AWS_ENDPOINT_URL_EVENTBRIDGE

serviceId	Cl id at de se pe il fil cc A\ **co**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
Evidently	ev	AWS_ENDPOINT_URL_EVIDENTLY
finspace	fi	AWS_ENDPOINT_URL_FINSPACE
finspace data	fi da	AWS_ENDPOINT_URL_FINSPACE_DATA
Firehose	fi	AWS_ENDPOINT_URL_FIREHOSE
fis	fi	AWS_ENDPOINT_URL_FIS
FMS	fr	AWS_ENDPOINT_URL_FMS
forecast	fo	AWS_ENDPOINT_URL_FORECAST
forecastquery	fo ue	AWS_ENDPOINT_URL_FORECASTQUERY
FraudDetector	fi ct	AWS_ENDPOINT_URL_FRAUDDETECTOR
FreeTier	fi	AWS_ENDPOINT_URL_FREETIER
FSx	fs	AWS_ENDPOINT_URL_FSX
GameLift	ga	AWS_ENDPOINT_URL_GAMELIFT

serviceId	Cl id at de se pe il fil cc A\ **ce**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
Glacier	gl	AWS_ENDPOINT_URL_GLACIER
Global Accelerator	gl ce	AWS_ENDPOINT_URL_GLOBAL_ACCELERATOR
Glue	gl	AWS_ENDPOINT_URL_GLUE
grafana	gr	AWS_ENDPOINT_URL_GRAFANA
Greengrass	gr s	AWS_ENDPOINT_URL_GREENGRASS
GreengrassV2	gr sv	AWS_ENDPOINT_URL_GREENGRASSV2
GroundStation	gr t:	AWS_ENDPOINT_URL_GROUNDSTATION
GuardDuty	gu	AWS_ENDPOINT_URL_GUARDDUTY
Health	he	AWS_ENDPOINT_URL_HEALTH
HealthLake	he e	AWS_ENDPOINT_URL_HEALTHLAKE

serviceId	Cl id at de se pe il fil cc A\ **cc**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
Honeycode	ho	AWS_ENDPOINT_URL_HONEYCODE
IAM	ia	AWS_ENDPOINT_URL_IAM
identitystore	id to	AWS_ENDPOINT_URL_IDENTITYSTORE
imagebuilder	im de	AWS_ENDPOINT_URL_IMAGEBUILDER
ImportExport	im oi	AWS_ENDPOINT_URL_IMPORTEXPORT
Inspector	in	AWS_ENDPOINT_URL_INSPECTOR
Inspector Scan	in _s	AWS_ENDPOINT_URL_INSPECTOR_SCAN
Inspector2	in 2	AWS_ENDPOINT_URL_INSPECTOR2
InternetMonitor	in or	AWS_ENDPOINT_URL_INTERNETMONITOR
IoT	io	AWS_ENDPOINT_URL_IOT

serviceId	Cl id at de se pe il fil cc A\ **cc**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
IoT Data Plane	ic p:	AWS_ENDPOINT_URL_IOT_DATA_PLANE
IoT Jobs Data Plane	ic da e	AWS_ENDPOINT_URL_IOT_JOBS_DATA_PLANE
IoT 1Click Devices Service	ic k_ _s	AWS_ENDPOINT_URL_IOT_1CLICK_DEVICES_ SERVICE
IoT 1Click Projects	ic k_ s	AWS_ENDPOINT_URL_IOT_1CLICK_PROJECTS
IoTAnalytics	ic ic	AWS_ENDPOINT_URL_IOTANALYTICS
IotDeviceAdvisor	ic ac	AWS_ENDPOINT_URL_IOTDEVICEADVISOR
IoT Events	ic s	AWS_ENDPOINT_URL_IOT_EVENTS
IoT Events Data	ic s_	AWS_ENDPOINT_URL_IOT_EVENTS_DATA

serviceId	Cl id at de se pe il fil cc A\ **co**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
IoTFleetHub	ic ul	AWS_ENDPOINT_URL_IOTFLEETHUB
IoTFleetWise	ic is	AWS_ENDPOINT_URL_IOTFLEETWISE
IoTSecureTunneling	ic tu	AWS_ENDPOINT_URL_IOTSECURETUNNELING
IoTSiteWise	ic se	AWS_ENDPOINT_URL_IOTSITEWISE
IoTThingsGraph	ic gr	AWS_ENDPOINT_URL_IOTTHINGSGRAPH
IoTTwinMaker	ic ke	AWS_ENDPOINT_URL_IOTTWINMAKER
IoT Wireless	ic es	AWS_ENDPOINT_URL_IOT_WIRELESS
ivs	iv	AWS_ENDPOINT_URL_IVS
IVS RealTime	iv ir	AWS_ENDPOINT_URL_IVS_REALTIME

serviceId	Cl id at de se pe il fil cc A\ **cc**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
ivschat	iv	AWS_ENDPOINT_URL_IVSCHAT
Kafka	ka	AWS_ENDPOINT_URL_KAFKA
KafkaConnect	ka ec	AWS_ENDPOINT_URL_KAFKACONNECT
kendra	ke	AWS_ENDPOINT_URL_KENDRA
Kendra Ranking	ke nl	AWS_ENDPOINT_URL_KENDRA_RANKING
Keyspaces	ke	AWS_ENDPOINT_URL_KEYSPACES
Kinesis	ki	AWS_ENDPOINT_URL_KINESIS
Kinesis Video Archived Media	ki id iv a	AWS_ENDPOINT_URL_KINESIS_VIDEO_ARCHI VED_MEDIA
Kinesis Video Media	ki id a	AWS_ENDPOINT_URL_KINESIS_VIDEO_MEDIA

serviceId	Cl id at de se pe il fil cc A\ **co**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
Kinesis Video Signaling	k: ic a:	AWS_ENDPOINT_URL_KINESIS_VIDEO_SIGNALING
Kinesis Video WebRTC Storage	k: ic t(e	AWS_ENDPOINT_URL_KINESIS_VIDEO_WEBRT C_STORAGE
Kinesis Analytics	k: na	AWS_ENDPOINT_URL_KINESIS_ANALYTICS
Kinesis Analytics V2	k: na v2	AWS_ENDPOINT_URL_KINESIS_ANALYTICS_V2
Kinesis Video	k: ic	AWS_ENDPOINT_URL_KINESIS_VIDEO
KMS	kr	AWS_ENDPOINT_URL_KMS
LakeFormation	la t:	AWS_ENDPOINT_URL_LAKEFORMATION
Lambda	la	AWS_ENDPOINT_URL_LAMBDA

serviceId	Cl id at de se pe il fil cc A\ **ce**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
Launch Wizard	la za	AWS_ENDPOINT_URL_LAUNCH_WIZARD
Lex Model Building Service	le _l _s	AWS_ENDPOINT_URL_LEX_MODEL_BUILDING_ SERVICE
Lex Runtime Service	le me e	AWS_ENDPOINT_URL_LEX_RUNTIME_SERVICE
Lex Models V2	le s_	AWS_ENDPOINT_URL_LEX_MODELS_V2
Lex Runtime V2	le me	AWS_ENDPOINT_URL_LEX_RUNTIME_V2
License Manager	li ar	AWS_ENDPOINT_URL_LICENSE_MANAGER
License Manager Linux Subscriptions	li ar nu r:	AWS_ENDPOINT_URL_LICENSE_MANAGER_LIN UX_SUBSCRIPTIONS

serviceId	Cl id at de se pe il fil cc A\ **cc**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
License Manager User Subscriptions	l: ar e: ip	AWS_ENDPOINT_URL_LICENSE_MANAGER_USER_SUBSCRIPTIONS
Lightsail	l:	AWS_ENDPOINT_URL_LIGHTSAIL
Location	lo	AWS_ENDPOINT_URL_LOCATION
CloudWatch Logs	c: h_	AWS_ENDPOINT_URL_CLOUDWATCH_LOGS
CloudWatch Logs	c: h_	AWS_ENDPOINT_URL_CLOUDWATCH_LOGS
LookoutEquipment	lo u:	AWS_ENDPOINT_URL_LOOKOUTEQUIPMENT
LookoutMetrics	lo t:	AWS_ENDPOINT_URL_LOOKOUTMETRICS
LookoutVision	lo s:	AWS_ENDPOINT_URL_LOOKOUTVISION
m2	m2	AWS_ENDPOINT_URL_M2

serviceId	Cl id at de se pe il fil cc A\ **ce**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
Machine Learning	ma ea	AWS_ENDPOINT_URL_MACHINE_LEARNING
Macie2	ma	AWS_ENDPOINT_URL_MACIE2
ManagedBlockchain	ma oc	AWS_ENDPOINT_URL_MANAGEDBLOCKCHAIN
ManagedBlockchain Query	ma oc qu	AWS_ENDPOINT_URL_MANAGEDBLOCKCHAIN_QUERY
Marketplace Agreement	ma ce er	AWS_ENDPOINT_URL_MARKETPLACE_AGREEMENT
Marketplace Catalog	ma ce g	AWS_ENDPOINT_URL_MARKETPLACE_CATALOG
Marketplace Deploymen t	ma ce me	AWS_ENDPOINT_URL_MARKETPLACE_DEPLOYMENT

serviceId	Cl id at de se pe il fil cc A\ **ce**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
Marketplace Entitleme nt Service	ma ce er v:	AWS_ENDPOINT_URL_MARKETPLACE_ENTITLE MENT_SERVICE
Marketplace Commerce Analytics	ma ce ce ic	AWS_ENDPOINT_URL_MARKETPLACE_COMMERC E_ANALYTICS
MediaConnect	me ec	AWS_ENDPOINT_URL_MEDIACONNECT
MediaConvert	me e:	AWS_ENDPOINT_URL_MEDIACONVERT
MediaLive	me	AWS_ENDPOINT_URL_MEDIALIVE
MediaPackage	me ag	AWS_ENDPOINT_URL_MEDIAPACKAGE
MediaPackage Vod	me ag	AWS_ENDPOINT_URL_MEDIAPACKAGE_VOD

serviceId	Cl id at de se pe il fil cc A\ **co**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
MediaPackageV2	me ag	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 ma	AWS_ENDPOINT_URL_MEDICAL_IMAGING
MemoryDB	me	AWS_ENDPOINT_URL_MEMORYDB
Marketplace Metering	ma ce ng	AWS_ENDPOINT_URL_MARKETPLACE_METERING
Migration Hub	m: _h	AWS_ENDPOINT_URL_MIGRATION_HUB
mgn	mg	AWS_ENDPOINT_URL_MGN

serviceId	Cl id at de se pe il fil cc A\ **cc**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
Migration Hub Refactor Spaces	m: _l ct es	AWS_ENDPOINT_URL_MIGRATION_HUB_REFAC TOR_SPACES
MigrationHub Config	m: hu g	AWS_ENDPOINT_URL_MIGRATIONHUB_CONFIG
MigrationHubOrches trator	m: hu t:	AWS_ENDPOINT_URL_MIGRATIONHUBORCHESTRATOR
MigrationHubStrategy	m: hu gy	AWS_ENDPOINT_URL_MIGRATIONHUBSTRATEGY
Mobile	mo	AWS_ENDPOINT_URL_MOBILE
mq	mo	AWS_ENDPOINT_URL_MQ
MTurk	mt	AWS_ENDPOINT_URL_MTURK
MWAA	mv	AWS_ENDPOINT_URL_MWAA
Neptune	ne	AWS_ENDPOINT_URL_NEPTUNE

serviceId	Cl id at de se pe il fil cc A\ **cc**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
Neptune Graph	ne ra	AWS_ENDPOINT_URL_NEPTUNE_GRAPH
neptunedata	ne ta	AWS_ENDPOINT_URL_NEPTUNEDATA
Network Firewall	ne i:	AWS_ENDPOINT_URL_NETWORK_FIREWALL
NetworkManager	ne na	AWS_ENDPOINT_URL_NETWORKMANAGER
NetworkMonitor	ne n:	AWS_ENDPOINT_URL_NETWORKMONITOR
nimble	n:	AWS_ENDPOINT_URL_NIMBLE
OAM	oa	AWS_ENDPOINT_URL_OAM
Omics	or	AWS_ENDPOINT_URL_OMICS
OpenSearch	op h	AWS_ENDPOINT_URL_OPENSEARCH
OpenSearchServerless	op hs ss	AWS_ENDPOINT_URL_OPENSEARCHSERVERLESS

serviceId	Cl id at de se pe il fil cc A\ **cc**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
OpsWorks	op	AWS_ENDPOINT_URL_OPSWORKS
OpsWorksCM	op m	AWS_ENDPOINT_URL_OPSWORKSCM
Organizations	or ic	AWS_ENDPOINT_URL_ORGANIZATIONS
OSIS	os	AWS_ENDPOINT_URL_OSIS
Outposts	ou	AWS_ENDPOINT_URL_OUTPOSTS
p8data	p8	AWS_ENDPOINT_URL_P8DATA
p8data	p8	AWS_ENDPOINT_URL_P8DATA
Panorama	pa	AWS_ENDPOINT_URL_PANORAMA
Payment Cryptography	pa ry hy	AWS_ENDPOINT_URL_PAYMENT_CRYPTOGRAPHY
Payment Cryptography Data	pa ry hy	AWS_ENDPOINT_URL_PAYMENT_CRYPTOGRAPHY_DATA

serviceId	Cl id at de se pe il fil cc A\ **co**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
Pca Connector Ad	pc ct	AWS_ENDPOINT_URL_PCA_CONNECTOR_AD
Personalize	pe ze	AWS_ENDPOINT_URL_PERSONALIZE
Personalize Events	pe ze	AWS_ENDPOINT_URL_PERSONALIZE_EVENTS
Personalize Runtime	pe ze e	AWS_ENDPOINT_URL_PERSONALIZE_RUNTIME
PI	pi	AWS_ENDPOINT_URL_PI
Pinpoint	pi	AWS_ENDPOINT_URL_PINPOINT
Pinpoint Email	pi er	AWS_ENDPOINT_URL_PINPOINT_EMAIL
Pinpoint SMS Voice	pi sr	AWS_ENDPOINT_URL_PINPOINT_SMS_VOICE

serviceId	Cl id at de se pe il fil cc A\ **cc**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
Pinpoint SMS Voice V2	p: sr _\	AWS_ENDPOINT_URL_PINPOINT_SMS_VOICE_V2
Pipes	p:	AWS_ENDPOINT_URL_PIPES
Polly	p(	AWS_ENDPOINT_URL_POLLY
Pricing	p:	AWS_ENDPOINT_URL_PRICING
PrivateNetworks	p: tv	AWS_ENDPOINT_URL_PRIVATENETWORKS
Proton	p:	AWS_ENDPOINT_URL_PROTON
QBusiness	ql	AWS_ENDPOINT_URL_QBUSINESS
QConnect	q(	AWS_ENDPOINT_URL_QCONNECT
QLDB	q:	AWS_ENDPOINT_URL_QLDB
QLDB Session	q: i(	AWS_ENDPOINT_URL_QLDB_SESSION
QuickSight	qu t	AWS_ENDPOINT_URL_QUICKSIGHT

serviceId	Cl id at de se pe il fil cc A\ **ce**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
RAM	ra	AWS_ENDPOINT_URL_RAM
rbin	rb	AWS_ENDPOINT_URL_RBIN
RDS	rd	AWS_ENDPOINT_URL_RDS
RDS Data	rd	AWS_ENDPOINT_URL_RDS_DATA
Redshift	re	AWS_ENDPOINT_URL_REDSHIFT
Redshift Data	re da	AWS_ENDPOINT_URL_REDSHIFT_DATA
Redshift Serverless	re se s	AWS_ENDPOINT_URL_REDSHIFT_SERVERLESS
Rekognition	re or	AWS_ENDPOINT_URL_REKOGNITION
repostspace	re ce	AWS_ENDPOINT_URL_REPOSTSPACE
resiliencehub	re el	AWS_ENDPOINT_URL_RESILIENCEHUB

serviceId	Cl id at de se pe il fil cc A\ **cc**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
Resource Explorer 2	re ex 2	AWS_ENDPOINT_URL_RESOURCE_EXPLORER_2
Resource Groups	re gr	AWS_ENDPOINT_URL_RESOURCE_GROUPS
Resource Groups Tagging API	re gr gg	AWS_ENDPOINT_URL_RESOURCE_GROUPS_TAG GING_API
RoboMaker	ro	AWS_ENDPOINT_URL_ROBOMAKER
RolesAnywhere	ro he	AWS_ENDPOINT_URL_ROLESANYWHERE
Route 53	ro	AWS_ENDPOINT_URL_ROUTE_53
Route53 Recovery Cluster	ro ec lu	AWS_ENDPOINT_URL_ROUTE53_RECOVERY_CLUSTER

serviceId	Cl id at de se pe il fil cc A\ **cc**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
Route53 Recovery Control Config	rc ec or nf	AWS_ENDPOINT_URL_ROUTE53_RECOVERY_CO NTROL_CONFIG
Route53 Recovery Readiness	rc ec ea	AWS_ENDPOINT_URL_ROUTE53_RECOVERY_RE ADINESS
Route 53 Domains	rc dc	AWS_ENDPOINT_URL_ROUTE_53_DOMAINS
Route53Resolver	rc sc	AWS_ENDPOINT_URL_ROUTE53RESOLVER
RUM	ru	AWS_ENDPOINT_URL_RUM
S3	s3	AWS_ENDPOINT_URL_S3
S3 Control	s3 l	AWS_ENDPOINT_URL_S3_CONTROL
S3Outposts	s3 s	AWS_ENDPOINT_URL_S3OUTPOSTS

serviceId	Cl id at de se pe il fil cc A\ **ce**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
SageMaker	sa	AWS_ENDPOINT_URL_SAGEMAKER
SageMaker A2I Runtime	sa _a ir	AWS_ENDPOINT_URL_SAGEMAKER_A2I_RUNTIME
Sagemaker Edge	sa _e	AWS_ENDPOINT_URL_SAGEMAKER_EDGE
SageMaker FeatureSt ore Runtime	sa _f to ir	AWS_ENDPOINT_URL_SAGEMAKER_FEATUREST ORE_RUNTIME
SageMaker Geospatial	sa _g a:	AWS_ENDPOINT_URL_SAGEMAKER_GEOSPATIAL
SageMaker Metrics	sa _m	AWS_ENDPOINT_URL_SAGEMAKER_METRICS
SageMaker Runtime	sa _r	AWS_ENDPOINT_URL_SAGEMAKER_RUNTIME

serviceId	Cl id at de se pe il fil cc A\ **cc**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
savingsplans	sa ar	AWS_ENDPOINT_URL_SAVINGSPLANS
Scheduler	sc	AWS_ENDPOINT_URL_SCHEDULER
schemas	sc	AWS_ENDPOINT_URL_SCHEMAS
SimpleDB	sd	AWS_ENDPOINT_URL_SIMPLEDB
Secrets Manager	se ar	AWS_ENDPOINT_URL_SECRETS_MANAGER
SecurityHub	se ub	AWS_ENDPOINT_URL_SECURITYHUB
SecurityLake	se al	AWS_ENDPOINT_URL_SECURITYLAKE
ServerlessApplicationRepository	se sa io to	AWS_ENDPOINT_URL_SERVERLESSAPPLICATIONREPOSITORY
Service Quotas	se uo	AWS_ENDPOINT_URL_SERVICE_QUOTAS

serviceId	Cl id at de se pe il fil cc A\ **ce**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
Service Catalog	se at	AWS_ENDPOINT_URL_SERVICE_CATALOG
Service Catalog AppRegistry	se at p:	AWS_ENDPOINT_URL_SERVICE_CATALOG_APP REGISTRY
ServiceDiscovery	se sc	AWS_ENDPOINT_URL_SERVICEDISCOVERY
SES	se	AWS_ENDPOINT_URL_SES
SESv2	se	AWS_ENDPOINT_URL_SESV2
Shield	sh	AWS_ENDPOINT_URL_SHIELD
signer	s:	AWS_ENDPOINT_URL_SIGNER
SimSpaceWeaver	s: ea	AWS_ENDPOINT_URL_SIMSPACEWEAVER
SMS	sr	AWS_ENDPOINT_URL_SMS
Snow Device Managemen t	sr ce me	AWS_ENDPOINT_URL_SNOW_DEVICE_MANAGEMENT

serviceId	Cl id at de se pe il fil cc A\ **cc**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
Snowball	sr	AWS_ENDPOINT_URL_SNOWBALL
SNS	sr	AWS_ENDPOINT_URL_SNS
SQS	sc	AWS_ENDPOINT_URL_SQS
SSM	ss	AWS_ENDPOINT_URL_SSM
SSM Contacts	ss ct	AWS_ENDPOINT_URL_SSM_CONTACTS
SSM Incidents	ss er	AWS_ENDPOINT_URL_SSM_INCIDENTS
Ssm Sap	ss	AWS_ENDPOINT_URL_SSM_SAP
SSO	ss	AWS_ENDPOINT_URL_SSO
SSO Admin	ss	AWS_ENDPOINT_URL_SSO_ADMIN
SSO OIDC	ss	AWS_ENDPOINT_URL_SSO_OIDC
SFN	st	AWS_ENDPOINT_URL_SFN
Storage Gateway	st at	AWS_ENDPOINT_URL_STORAGE_GATEWAY

serviceId	Cl id at de se pe il fil cc A\ **cc**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
STS	st	AWS_ENDPOINT_URL_STS
SupplyChain	su in	AWS_ENDPOINT_URL_SUPPLYCHAIN
Support	su	AWS_ENDPOINT_URL_SUPPORT
Support App	su pp	AWS_ENDPOINT_URL_SUPPORT_APP
SWF	sv	AWS_ENDPOINT_URL_SWF
synthetics	sy s	AWS_ENDPOINT_URL_SYNTHETICS
Textract	te	AWS_ENDPOINT_URL_TEXTRACT
Timestream InfluxDB	ti m_ b	AWS_ENDPOINT_URL_TIMESTREAM_INFLUXDB
Timestream Query	ti m_	AWS_ENDPOINT_URL_TIMESTREAM_QUERY
Timestream Write	ti m_	AWS_ENDPOINT_URL_TIMESTREAM_WRITE

serviceId	Cl id at de se pe il fil cc A\ **ce**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
tnb	tr	AWS_ENDPOINT_URL_TNB
Transcribe	t: e	AWS_ENDPOINT_URL_TRANSCRIBE
Transfer	t:	AWS_ENDPOINT_URL_TRANSFER
Translate	t:	AWS_ENDPOINT_URL_TRANSLATE
TrustedAdvisor	t: v:	AWS_ENDPOINT_URL_TRUSTEDADVISOR
VerifiedPermissions	ve e: s	AWS_ENDPOINT_URL_VERIFIEDPERMISSIONS
Voice ID	vo	AWS_ENDPOINT_URL_VOICE_ID
VPC Lattice	vp ce	AWS_ENDPOINT_URL_VPC_LATTICE
WAF	wa	AWS_ENDPOINT_URL_WAF
WAF Regional	wa na	AWS_ENDPOINT_URL_WAF_REGIONAL

serviceId	Cl id at de se pe il fil cc A\ **cc**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
WAFV2	wa	AWS_ENDPOINT_URL_WAFV2
WellArchitected	we te	AWS_ENDPOINT_URL_WELLARCHITECTED
Wisdom	wi	AWS_ENDPOINT_URL_WISDOM
WorkDocs	wo	AWS_ENDPOINT_URL_WORKDOCS
WorkLink	wo	AWS_ENDPOINT_URL_WORKLINK
WorkMail	wo	AWS_ENDPOINT_URL_WORKMAIL
WorkMailMessageFlow	wo es w	AWS_ENDPOINT_URL_WORKMAILMESSAGEFLOW
WorkSpaces	wo s	AWS_ENDPOINT_URL_WORKSPACES
WorkSpaces Thin Client	wo s_ ie	AWS_ENDPOINT_URL_WORKSPACES_THIN_CLIENT
WorkSpaces Web	wo s_	AWS_ENDPOINT_URL_WORKSPACES_WEB

serviceId	Cl id at de se pe il fil co A\ **co**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
XRay	x:	AWS_ENDPOINT_URL_XRAY

Impostazioni predefinite di configurazione intelligente

Con la funzionalità Smart Configuration Defaults, AWS gli SDK possono fornire valori predefiniti e ottimizzati per altre impostazioni di configurazione.

Configura questa funzionalità utilizzando quanto segue:

defaults_mode- impostazione dei AWS **config** file condivisi, **AWS_DEFAULTS_MODE**- variabile d'ambiente, **aws.defaultsMode**- Proprietà del sistema JVM: solo Java/Kotlin

Con questa impostazione, puoi scegliere una modalità che si allinea all'architettura dell'applicazione, che fornisce quindi valori predefiniti ottimizzati per l'applicazione. Se un'impostazione AWS SDK ha un valore impostato in modo esplicito, quel valore ha sempre la precedenza. Se un'impostazione AWS SDK non ha un valore impostato in modo esplicito e non `defaults_mode` è uguale a quella precedente, questa funzionalità può fornire valori predefiniti diversi per varie impostazioni ottimizzate per l'applicazione. Le impostazioni possono includere quanto segue: impostazioni di comunicazione HTTP, comportamento dei tentativi, impostazioni regionali degli endpoint del servizio e, potenzialmente, qualsiasi configurazione relativa all'SDK. I clienti che utilizzano questa funzionalità possono ottenere nuove impostazioni di configurazione predefinite personalizzate per scenari di utilizzo comuni. Se il tuo non `defaults_mode` è uguale `alegacy`, ti consigliamo di eseguire dei test dell'applicazione quando aggiorni l'SDK, poiché i valori predefiniti forniti potrebbero cambiare man mano che le best practice evolvono.

Valore predefinito: `legacy`

Nota: per impostazione predefinita, le nuove versioni principali degli SDK saranno impostate su. `standard`

Valori validi:

- `legacy`— Fornisce impostazioni predefinite che variano in base all'SDK ed esistevano prima della creazione di. `defaults_mode`
- `standard`— Fornisce i valori predefiniti più recenti consigliati che dovrebbero essere sicuri per l'esecuzione nella maggior parte degli scenari.
- `in-region`— Si basa sulla modalità standard e include un'ottimizzazione personalizzata per le applicazioni che effettuano chiamate Servizi AWS dall'interno della stessa Regione AWS.
- `cross-region`— Si basa sulla modalità standard e include un'ottimizzazione personalizzata per le applicazioni che effettuano chiamate Servizi AWS in una regione diversa.
- `mobile`— Si basa sulla modalità standard e include un'ottimizzazione personalizzata per le applicazioni mobili.
- `auto`— Si basa sulla modalità standard e include funzionalità sperimentali. L'SDK tenta di scoprire l'ambiente di runtime per determinare automaticamente le impostazioni appropriate. Il rilevamento automatico è basato sull'euristica e non fornisce una precisione del 100%. Se non è possibile determinare l'ambiente di esecuzione, `standard` viene utilizzata la modalità. Il rilevamento automatico potrebbe interrogare i metadati dell'istanza e i dati utente, il che potrebbe introdurre latenza. Se la latenza di avvio è fondamentale per la tua applicazione, ti consigliamo invece di sceglierne una esplicita. `defaults_mode`

Esempio di impostazione di questo valore nel `config` file:

```
[default]
defaults_mode = standard
```

I seguenti parametri potrebbero essere ottimizzati in base alla selezione di`defaults_mode`:

- `retryMode`— specifica in che modo l'SDK tenta di riprovare. Per informazioni, consulta Comportamento di ripetizione.
- `stsRegionalEndpoints`— Specifica in che modo l'SDK determina l' Servizio AWS endpoint che utilizza per comunicare con (). AWS Security Token Service AWS STS Per informazioni, consulta AWS STS Endpoint regionalizzati.
- `s3UsEast1RegionalEndpoints`: specifica in che modo l'SDK determina l'endpoint del AWS servizio che utilizza per comunicare con Amazon S3 per la regione. `us-east-1`

- `connectTimeoutInMillis`— Dopo aver effettuato un tentativo di connessione iniziale su un socket, il periodo di tempo prima del timeout. Se il client non riceve il completamento dell'handshake di connessione, rinuncia e fallisce l'operazione.
- `tlsNegotiationTimeoutInMillis`— Il tempo massimo che un handshake TLS può impiegare dal momento in cui il messaggio CLIENT HELLO viene inviato al momento in cui il client e il server hanno completamente negoziato i codici e si sono scambiati le chiavi.

Il valore predefinito per ogni impostazione cambia a seconda di quella selezionata per l'applicazione. `defaults_mode` Questi valori sono attualmente impostati come segue (soggetti a modifiche):

Parametro	Modalità **`standard`**	Modalità **`in-region`**	Modalità **`cross-region`**	Modalità **`mobile`**
`retryMode`	`standard`	`standard`	`standard`	`standard`
`stsRegionalEndpoints`	`regional`	`regional`	`regional`	`regional`
`s3UsEast1RegionalEndpoints`	`regional`	`regional`	`regional`	`regional`
`connectTimeoutInMillis`	3100	1100	3100	30000
`tlsNegotiationTimeoutInMillis`	3100	1100	3100	30000

Ad esempio, `defaults_mode` se l'opzione selezionata fosse`standard`, il valore di `standard` verrebbe assegnato a `retry_mode` (dalle `retry_mode` opzioni valide) e il valore di `regional` verrebbe assegnato a `stsRegionalEndpoints` (dalle `stsRegionalEndpoints` opzioni valide).

Compatibilità con gli AWS SDK

I seguenti SDK supportano le funzionalità e le impostazioni descritte in questo argomento. Vengono annotate eventuali eccezioni parziali. Tutte le impostazioni delle proprietà del sistema JVM sono supportate solo da AWS SDK for Java and the. SDK AWS for Kotlin

SDK	Supportato	Note o ulteriori informazioni
AWS CLI v2	No	
SDK per C++	Sì	Parametri non ottimizzati:`stsRegionalEndpoints` ,,`s3UsEast1RegionalEndpoints` . `tlsNegotiationTimeoutInMillis`
SDK per Go V2 (1.x)	Sì	Parametri non ottimizzati:`retryMode` ,,. `stsRegionalEndpoints` `s3UsEast1RegionalEndpoints`
SDK per Go 1.x (V1)	No	
SDK per Java 2.x	Sì	Parametri non ottimizzati:. `stsRegionalEndpoints`
SDK per Java 1.x	No	
SDK per 3.x JavaScript	Sì	Parametri non ottimizzati:`stsRegionalEndpoints` ,,`s3UsEast1RegionalEndpoints` . `tlsNegotiationTimeoutInMillis` `connectTimeoutInMillis` viene chiamato`connectionTimeout` .

SDK	Supportato	Note o ulteriori informazioni
SDK per 2.x JavaScript	No	
SDK per Kotlin	No	
SDK per.NET 3.x	Sì	Parametri non ottimizza ti:`connectTimeoutInMillis` ,. `tlsNegotiationTimeoutInMillis`
SDK per PHP 3.x	Sì	Parametri non ottimizzati:. `tlsNegotiationTimeoutInMillis`
SDK per Python (Boto3)	Sì	Parametri non ottimizzati:. `tlsNegotiationTimeoutInMillis`
SDK per Ruby 3.x	Sì	
SDK per Rust	No	
Strumenti per PowerShell	Sì	Parametri non ottimizza ti:`connectTimeoutInMillis` ,`tlsNegotiationTimeoutInMillis` .

AWSLibrerie Common Runtime (CRT)

Le librerie AWS Common Runtime (CRT) sono una libreria di base degli SDK. Il CRT è una famiglia modulare di pacchetti indipendenti, scritta in C. Ogni pacchetto offre buone prestazioni e un ingombro minimo per le diverse funzionalità richieste. Queste funzionalità sono comuni e condivise tra tutti gli SDK e offrono un migliore riutilizzo, ottimizzazione e precisione del codice. I pacchetti sono:

- `awslabs/aws-c-auth`: autenticazione AWS lato client (provider di credenziali standard e firma (sigv4))
- `awslabs/aws-c-cal`: tipi primitivi crittografici, hash (MD5, SHA256, SHA256 HMAC), firmatari, AES
- `awslabs/aws-c-common`: Strutture dati di base, tipi primitivi di threading/sincronizzazione, gestione del buffer, funzioni relative a stdlib
- `awslabs/aws-c-compression`: Algoritmi di compressione (codifica/decodifica Huffman)
- `awslabs/aws-c-event-stream`: elaborazione dei messaggi di flusso di eventi (headers, prelude, payload, crc/trailer), implementazione di chiamate di procedura remote (RPC) su flussi di eventi
- `awslabs/aws-c-http`: implementazione in C99 delle specifiche HTTP/1.1 e HTTP/2
- `awslabs/aws-c-io`: Socket (TCP, UDP), DNS, pipe, loop di eventi, canali, SSL/TLS
- `awslabs/aws-c-iot`: implementazione C99 dell'integrazione dei servizi cloud AWS IoT con i dispositivi
- `awslabs/aws-c-mqtt`: Protocollo di messaggistica standard e leggero per l'Internet of Things (IoT)
- `awslabs/aws-c-s3`: implementazione della libreria C99 per la comunicazione con il servizio Amazon S3, progettata per massimizzare il throughput su istanze Amazon EC2 a larghezza di banda elevata
- `awslabs/aws-c-sdkutils`: Una libreria di utilità per l'analisi e la gestione dei profili AWS
- `awslabs/aws-checksums`: CRC32c e CRC32 multipiattaforma con accelerazione hardware con riserva di implementazioni software efficienti
- `awslabs/aws-lc`: libreria crittografica generica gestita dal team di AWS crittografia AWS e dai suoi clienti, basata sul codice del progetto Google BoringSSL e del progetto OpenSSL
- `awslabs/s2n`: Implementazione C99 dei protocolli TLS/SSL, progettata per essere piccola e veloce con la sicurezza come priorità

Il CRT è disponibile tramite tutti gli SDK tranne Go.

Dipendenze CRT

Le librerie CRT formano una rete complessa di relazioni e dipendenze. Conoscere queste relazioni è utile se è necessario creare il CRT direttamente dal codice sorgente. Tuttavia, la maggior parte degli utenti accede alla funzionalità CRT tramite l'SDK del linguaggio (come AWS SDK per C++ o SDK AWS per Java) o l'SDK del dispositivo IoT del linguaggio (come IoT SDK per C++ o IoT SDK AWS per Java). AWS Nel diagramma seguente, la casella Language CRT Bindings si riferisce al pacchetto che include le librerie CRT per un SDK linguistico specifico. Questa è una raccolta di pacchetti del modulo`aws-crt-*`, dove '*' è un linguaggio SDK (come o). `aws-crt-cpp`aws-crt-java

Di seguito è riportata un'illustrazione delle dipendenze gerarchiche delle librerie CRT.

AWS Politica di manutenzione degli SDK e degli strumenti

Panoramica

Questo documento delinea la politica di manutenzione per i kit e gli strumenti di sviluppo AWS software (SDK), inclusi gli SDK per dispositivi mobili e IoT, e le relative dipendenze sottostanti. AWS fornisce regolarmente agli AWS SDK e agli strumenti aggiornamenti che possono contenere supporto per AWS API nuove o aggiornate, nuove funzionalità, miglioramenti, correzioni di bug, patch di sicurezza o aggiornamenti della documentazione. Gli aggiornamenti possono anche riguardare le modifiche relative alle dipendenze, ai runtime delle lingue e ai sistemi operativi. AWS Le versioni SDK vengono pubblicate nei gestori di pacchetti (ad esempio Maven, NuGet PyPI) e sono disponibili come codice sorgente su. GitHub

Consigliamo agli utenti di rimanere up-to-date con le versioni SDK per tenersi aggiornati sulle funzionalità più recenti, sugli aggiornamenti di sicurezza e sulle dipendenze sottostanti. L'uso continuato di una versione SDK non supportata non è consigliato e viene eseguito a discrezione dell'utente.

Controllo delle versioni

Le versioni di rilascio dell' AWS SDK sono in formato X.Y.Z dove X rappresenta la versione principale. L'aumento della versione principale di un SDK indica che questo SDK ha subito modifiche significative e sostanziali per supportare nuovi idiomi e modelli nel linguaggio. Le versioni principali vengono introdotte quando le interfacce pubbliche (ad esempio classi, metodi, tipi, ecc.), i comportamenti o la semantica sono cambiati. Le applicazioni devono essere aggiornate per poter funzionare con la versione SDK più recente. È importante aggiornare le versioni principali con attenzione e in conformità con le linee guida per l'aggiornamento fornite da. AWS

Ciclo di vita della versione principale dell'SDK

Il ciclo di vita delle principali versioni di SDK e Tools è costituito da 5 fasi, descritte di seguito.

- Anteprima per sviluppatori (Fase 0): durante questa fase, gli SDK non sono supportati, non devono essere utilizzati negli ambienti di produzione e sono pensati solo per l'accesso anticipato e il feedback. È possibile che le versioni future introducano modifiche sostanziali. Una volta AWS identificata una versione come prodotto stabile, può contrassegnarla come Release Candidate. Le

Release Candidate sono pronte per la versione GA, a meno che non emergano bug significativi, e riceveranno un supporto completo AWS .

- Disponibilità generale (GA) (Fase 1): durante questa fase, gli SDK sono completamente supportati. AWS fornirà versioni SDK regolari che includono il supporto per nuovi servizi, aggiornamenti delle API per i servizi esistenti e correzioni di bug e sicurezza. Per Tools, AWS fornirà versioni regolari che includono nuovi aggiornamenti delle funzionalità e correzioni di bug. AWS supporterà la versione GA di un SDK per almeno 24 mesi.
- Annuncio di manutenzione (Fase 2): AWS pubblicherà un annuncio pubblico almeno 6 mesi prima che un SDK entri in modalità di manutenzione. Durante questo periodo, l'SDK continuerà a essere completamente supportato. In genere, la modalità di manutenzione viene annunciata contemporaneamente al passaggio della versione principale successiva a GA.
- Manutenzione (Fase 3): durante la modalità di manutenzione, AWS limita le versioni SDK per risolvere solo le correzioni di bug e i problemi di sicurezza critici. Un SDK non riceverà aggiornamenti delle API per servizi nuovi o esistenti né verrà aggiornato per supportare nuove regioni. La modalità di manutenzione ha una durata predefinita di 12 mesi, se non diversamente specificato.
- Fine del supporto (Fase 4): quando un SDK raggiunge la fine del supporto, non riceverà più aggiornamenti o versioni. Le versioni pubblicate in precedenza continueranno a essere disponibili tramite gestori di pacchetti pubblici e il codice rimarrà attivo. GitHub Il GitHub repository può essere archiviato. L'uso di un SDK raggiunto end-of-support viene effettuato a discrezione dell'utente. Consigliamo agli utenti di eseguire l'aggiornamento alla nuova versione principale.

Di seguito è riportata un'illustrazione visiva del ciclo di vita della versione principale dell'SDK. Tieni presente che le tempistiche riportate di seguito sono illustrative e non vincolanti.

Ciclo di vita delle dipendenze

La maggior parte degli AWS SDK ha dipendenze sottostanti, come i runtime del linguaggio, i sistemi operativi o le librerie e i framework di terze parti. Queste dipendenze sono in genere legate alla comunità linguistica o al fornitore proprietario di quel particolare componente. Ogni comunità o fornitore pubblica la propria end-of-support pianificazione per il proprio prodotto.

I seguenti termini vengono utilizzati per classificare le dipendenze sottostanti di terze parti:

- Sistema operativo (OS): alcuni esempi includono Amazon Linux AMI, Amazon Linux 2, Windows 2008, Windows 2012, Windows 2016, ecc.

- Language Runtime: gli esempi includono Java 7, Java 8, Java 11, .NET Core, .NET Standard, .NET PCL, ecc.
- Libreria/Framework di terze parti: gli esempi includono OpenSSL, .NET Framework 4.5, Java EE, ecc.

La nostra politica prevede di continuare a supportare le dipendenze SDK per almeno 6 mesi dopo la fine del supporto per la dipendenza da parte della community o del fornitore. Questa politica, tuttavia, potrebbe variare a seconda della dipendenza specifica.

> **Note**
>
> AWS si riserva il diritto di interrompere il supporto per una dipendenza sottostante senza aumentare la versione principale dell'SDK

Metodi di comunicazione

Gli annunci di manutenzione vengono comunicati in diversi modi:

- Agli account interessati viene inviato un annuncio via e-mail che annuncia i nostri piani per terminare il supporto per la versione SDK specifica. L'e-mail illustrerà il percorso da seguire end-of-support, specificherà le tempistiche della campagna e fornirà indicazioni per l'aggiornamento.
- AWS La documentazione SDK, come la documentazione di riferimento sulle API, le guide per l'utente, le pagine di marketing dei prodotti SDK e i GitHub readme, viene aggiornata per indicare la tempistica della campagna e fornire indicazioni sull'aggiornamento delle applicazioni interessate.
- Viene pubblicato un post AWS sul blog che delinea il percorso e ribadisce le tempistiche della end-of-support campagna.
- Gli avvisi di obsolescenza vengono aggiunti agli SDK, delineando il percorso e il collegamento alla documentazione SDK. end-of-support

Per visualizzare l'elenco delle versioni principali disponibili di AWS SDK e strumenti e la relativa fase del ciclo di manutenzione, consulta. Matrice di supporto delle versioni

AWS Matrice di supporto delle versioni degli SDK e degli strumenti

La matrice seguente mostra l'elenco delle versioni principali del AWS Software Development Kit (SDK) disponibili e la loro posizione nel ciclo di vita della manutenzione con le relative tempistiche. Per informazioni dettagliate sul ciclo di vita delle versioni principali di AWS SDK e strumenti e sulle relative dipendenze sottostanti, consulta. Politica di manutenzione

SDK	Versione principale	Fase attuale	Data di disponibilità generale	Note
AWS CLI	1.x	Disponibilità generale	2/09/2013	
AWS CLI	2.x	Disponibilità generale	10/2/2020	
SDK per C++	1.x	Disponibilità generale	2/09/2015	
SDK per Go V2	Versione 2 1.x	Disponibilità generale	19/01/2021	
SDK for Go	1.x	Annuncio di manutenzione	19/11/2015	Vedi l'annuncio per dettagli e date
SDK per Java	1.x	Annuncio di manutenzione	25/03/2010	Vedi l'annuncio per dettagli e date
SDK per Java	2.x	Disponibilità generale	20/11/2018	
SDK per JavaScript	1.x	Fine del supporto	6/05/2013	

SDK	Versione principale	Fase attuale	Data di disponibilità generale	Note
SDK per JavaScript	2.x	Annuncio di manutenzione	19/06/2014	Vedi l'annuncio per dettagli e date
SDK per JavaScript	3.x	Disponibilità generale	15/12/2020	
SDK per Kotlin	1.x	Disponibilità generale	27/11/2023	
SDK per.NET	1.x	Fine del supporto	11/2009	
SDK per.NET	2.x	Fine del supporto	8/11/2013	
SDK per.NET	3.x	Disponibilità generale	28/07/2015	
SDK per PHP	2.x	Fine del supporto	02/11/2012	
SDK per PHP	3.x	Disponibilità generale	27/05/2015	
SDK per Python (Boto2)	1.x	Fine del supporto	13/07/2011	
SDK per Python (Boto3)	1.x	Disponibilità generale	22/06/2015	
SDK per Python (Botocore)	1.x	Disponibilità generale	22/06/2015	
SDK per Ruby	1.x	Fine del supporto	14/07/2011	

SDK	Versione principale	Fase attuale	Data di disponibilità generale	Note
SDK per Ruby	2.x	Fine del supporto	15/02/2015	
SDK per Ruby	3.x	Disponibilità generale	29/08/2017	
SDK per Rust	1.x	Disponibilità generale	27/11/2023	
SDK per Swift	1.x	Anteprima per sviluppatori		
Strumenti per PowerShell	2.x	Fine del supporto	8/11/2013	
Utensili per PowerShell	3.x	Fine del supporto	29/07/2015	
Utensili per PowerShell	4.x	Disponibilità generale	21/11/2019	

Guida di riferimento alla cronologia dei documenti per AWS SDK e strumenti

La tabella seguente descrive importanti aggiunte e aggiornamenti alla Guida di riferimento agli AWS SDK e agli strumenti. Per ricevere notifiche sugli aggiornamenti della documentazione, puoi sottoscrivere il feed RSS.

Modifica	Descrizione	Data
Proprietà di sistema SDK for Java 1.x	Aggiungi dettagli sulle impostazioni di configurazione del sistema JVM supportate dalla versione 1.x. AWS SDK for Java	30 maggio 2024
Aggiornamenti delle impostazioni	Aggiungere le impostazioni di configurazione del sistema JVM.	27 marzo 2024
Aggiornamenti della tabella di compatibilità	Aggiornamenti alla compatibilità per il supporto SDK, aggiornamenti alle procedure di IAM Identity Center.	20 febbraio 2024
Aggiornamento delle credenziali del contenitore. Aggiornamento IMDS.	Aggiunta del supporto per Amazon EKS. Aggiunta un'impostazione per disabilitare il fallback di IMDSv1.	29 dicembre 2023
Richiedi la compressione	Aggiungere impostazioni per la funzionalità di compressione delle richieste.	27 dicembre 2023
Tabelle di compatibilità	Tabelle di compatibilità per SDK e funzionalità degli strumenti aggiornate per includere SDK per Kotlin, SDK	10 dicembre 2023

	per Rust e. AWS Tools for PowerShell	
Aggiornamenti di autenticazione	Aggiornamenti ai metodi di autenticazione supportati per SDK e strumenti.	1 luglio 2023
Aggiornamenti delle best practice di IAM	Guida aggiornata per l'allineamento alle best practice IAM. Per ulteriori informazioni, consulta Best practice per la sicurezza in IAM.	27 febbraio 2023
Aggiornamenti SSO	Aggiornamenti alle credenziali SSO per la nuova configurazione del token SSO.	19 novembre 2022
Aggiornamenti delle impostazioni	Aggiornamenti alla tabella di supporto per la configurazione generale e per i punti di accesso multiregionali di Amazon S3.	17 novembre 2022
Aggiornamenti delle impostazioni	Aggiornamenti alla chiarezza del client IMDS e delle credenziali IMDS. Aggiornamenti alle variabili di ambiente.	4 novembre 2022
Aggiornamento della pagina di benvenuto	Annuncio di Amazon CodeWhisperer.	22 settembre 2022
Modifica del nome del servizio per Single Sign-On	Aggiornamenti che riflettono il fatto che l' AWS SSO viene ora denominato. AWS IAM Identity Center	26 luglio 2022

Aggiornamento delle impostazioni	Aggiornamenti minori ai dettagli del file di configurazione e alle impostazioni supportate.	15 giugno 2022
Aggiorna	Aggiornamento massiccio di quasi tutte le parti di questa guida.	1 febbraio 2022
Versione iniziale	La prima versione di questa guida viene rilasciata al pubblico.	13 marzo 2020

Glossario AWS

Per la terminologia AWS più recente, consultare il glossario AWS nella documentazione di riferimento per Glossario AWS.

Le traduzioni sono generate tramite traduzione automatica. In caso di conflitto tra il contenuto di una traduzione e la versione originale in Inglese, quest'ultima prevarrà.