Guida di riferimento

AWS SDKs e strumenti

AWS SDKs e strumenti: Guida di riferimento

Table of Contents

Cosa è trattato nella Guida di riferimento agli strumenti AWS SDKs e strumenti

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

Questa guida include informazioni relative a:

- Configurazione AWS SDKs e strumenti a livello globale— Come utilizzare le variabili shared `config` e `credentials` files o di ambiente per configurare AWS SDKs i propri strumenti.
- Autenticazione e accesso tramite strumenti AWS SDKs e— Stabilisci in che modo il codice o lo strumento si autentica AWS durante lo sviluppo con. Servizi AWS
- AWS SDKs riferimento alle impostazioni e agli strumenti— Riferimento per tutte le impostazioni standardizzate disponibili per l'autenticazione e la configurazione.
- AWS Librerie Common Runtime (CRT)— Panoramica delle librerie AWS Common Runtime (CRT) condivise disponibili per quasi tutti. SDKs
- AWS SDKs e politica di manutenzione degli strumenticopre la politica di manutenzione e il controllo delle versioni per i kit di sviluppo AWS software (SDKs) e gli strumenti, inclusi dispositivi mobili e Internet of Things (IoT) SDKs, e le loro dipendenze sottostanti.

Questa guida di riferimento AWS SDKs e agli strumenti intende essere una base di informazioni applicabile a più SDKs 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 SDKs e politica di manutenzione degli strumenti
• AWS Cloud9 • AWS CDK • AWS Toolkit per Azure DevOps • AWS Toolkit for JetBrains	Configurazione AWS SDKs e strumenti a livello globale

Se stai usando:	Le sezioni pertinenti di questa guida per te sono:
• AWS Toolkit for Visual Studio • AWS Toolkit for Visual Studio Code • AWS Serverless Application Model • AWS CodeArtifact • AWS CodeBuild • Amazon CodeCatalyst • AWS CodeCommit • AWS CodeDeploy • AWS CodePipeline	Autenticazione e accesso tramite strumenti AWS SDKs e AWS SDKs e politica di manutenzione degli strumenti
• AWS CLI • AWS SDK per C++ • AWS SDK per Go • AWS SDK per Java • AWS SDK per JavaScript • AWS SDK per Kotlin • AWS SDK per .NET • AWS SDK per PHP • AWS SDK per Python (Boto3) • AWS SDK per Ruby • AWS SDK per Rust • AWS SDK per Swift • AWS Tools for Windows PowerShell	Configurazione AWS SDKs e strumenti a livello globale Autenticazione e accesso tramite strumenti AWS SDKs e AWS SDKs riferimento alle impostazioni e agli strumenti AWS Librerie Common Runtime (CRT) AWS SDKs e politica di manutenzione degli strumenti AWS SDKs ciclo di vita delle versioni e degli strumenti

- 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.
- Per la AWS terminologia, consulta il AWS glossario nella Guida di riferimento.Glossario AWS

Risorse per gli sviluppatori

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 Guida per l'utente di Amazon Q Developer.

Notifica telemetrica del Toolkit

AWS I toolkit IDE (Integrated Development Environment) sono plugin ed estensioni che consentono l'accesso ai servizi dell'IDE. AWS I plugin e le estensioni IDE di Amazon Q consentono l'assistenza AI generativa nel tuo IDE. Per informazioni dettagliate su ciascuno dei Toolkit IDE, consulta le Guide per l'utente del Toolkit nella tabella precedente. Per ulteriori informazioni sull'utilizzo di Amazon Q nel tuo IDE, consulta l'argomento Uso di Amazon Q nell'IDE nella guida per sviluppatori di Amazon Q.

AWS IDE Toolkits e Amazon Q possono raccogliere e archiviare dati di telemetria lato client per prendere decisioni sulle future versioni di Toolkit e Amazon Q. AWS I dati raccolti quantificano l'utilizzo del AWS Toolkit e di Amazon Q.

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

Per informazioni dettagliate sui dati di telemetria raccolti da ciascuno degli AWS IDE Toolkit e delle estensioni Amazon Q, consulta i documenti delle risorse nei seguenti repository Toolkit: AWS GitHub

- AWS Toolkit di Visual Studio con Amazon Q
- AWS Toolkit for Visual Studio Code e l'estensione Amazon Q per VS Code
- AWS Toolkit for JetBrains e il plugin Amazon Q per JetBrains
- Amazon Q per Eclipse

Alcuni AWS servizi accessibili nei AWS Toolkit possono raccogliere dati di telemetria aggiuntivi sul lato client. 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 AWS SDKs e strumenti a livello globale

Con AWS SDKs e altri strumenti per AWS sviluppatori, come AWS Command Line Interface (AWS CLI), puoi interagire con il AWS servizio APIs. 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, gli strumenti AWS SDKs e supportano impostazioni comuni provenienti da fonti primarie oltre al codice stesso. Questa sezione comprende i seguenti argomenti:

Argomenti

Utilizzo di **`credentials`** file **`config`** e file condivisi per configurare AWS SDKs e utilizzare strumenti a livello globale

I `credentials` file AWS `config` condivisi sono il modo più comune per specificare l'autenticazione e la configurazione su un AWS SDK o uno strumento.

I `credentials` file condivisi `config` e contengono un set di profili. Un profilo è un insieme di impostazioni di configurazione, in coppie chiave-valore, utilizzato da AWS SDKs, the AWS Command Line Interface (AWS CLI) e altri strumenti. I valori di configurazione sono allegati a un profilo per configurare alcuni aspetti del SDK/tool momento in cui tale profilo viene utilizzato. Questi file sono «condivisi» in quanto i valori hanno effetto su qualsiasi applicazione, processo o SDKs sull'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.

> **Note**
>
> Se l'applicazione si trova su un server che esegue più applicazioni, si consiglia di utilizzare sempre profili denominati anziché il profilo predefinito. Il profilo predefinito viene selezionato automaticamente da qualsiasi AWS applicazione nell'ambiente e viene condiviso tra di esse. Pertanto, se qualcun altro aggiorna il profilo predefinito per la propria applicazione, ciò può influire involontariamente sugli altri. Per evitare ciò, definite un profilo denominato nel `config` file condiviso e quindi utilizzate quel profilo denominato nell'applicazione impostando il profilo

denominato nel codice. Puoi utilizzare la variabile di ambiente o la proprietà di sistema JVM per impostare il profilo denominato se sai che l'ambito influisce solo sulla tua applicazione.

Formato del file di configurazione

Il `config` file è organizzato in sezioni. Una sezione è una raccolta denominata di impostazioni e prosegue fino a quando non si incontra 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 il formato generale `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 su 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'`s3`impostazione 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. Utilizzo di IAM Identity Center per autenticare AWS SDK e strumenti Un profilo è collegato a una `sso-session` sezione da 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 strumenti che SDKs includono questa funzionalità. Per vedere se questa funzionalità è disponibile per il tuo SDK, consulta per gli endpoint specifici del servizio. Support by AWS SDKs and tools

La riga di definizione della `services` sezione indica un gruppo di impostazioni che configurano 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 `services`, consulta Identificatori per endpoint specifici del servizio. La chiave identificativa del servizio è seguita da impostazioni annidate, ciascuna sulla propria riga e con un rientro di 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 con rientro immediatamente successive vengono 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.

Individuazione e modifica della posizione delle risorse condivise **`config`** e dei **`credentials`** file AWS SDKs e degli strumenti

I file condivisi AWS `config` e sono `credentials` file di testo semplice che contengono informazioni di configurazione per gli AWS SDKs strumenti e. 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. Utilizzo di variabili di ambiente per configurare AWS SDKs e utilizzare strumenti a livello globale

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.

Utilizzo di variabili di ambiente per configurare AWS SDKs e utilizzare strumenti a livello globale

Le variabili di ambiente forniscono un altro modo per specificare le opzioni di configurazione e le credenziali durante l'utilizzo AWS SDKs degli strumenti. Le variabili di ambiente possono essere utili per creare script o impostare temporaneamente un profilo denominato come predefinito. Per l'elenco delle variabili di ambiente supportate dalla maggior parte SDKs, vedereelenco 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` `ecredentials`.
- 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, con AWS 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.

Utilizzo delle proprietà del sistema JVM per configurare globalmente e AWS SDK per Java AWS SDK per Kotlin

Le proprietà del sistema JVM forniscono un altro modo per specificare le opzioni di configurazione e le credenziali per SDKs l'esecuzione sulla JVM, ad esempio la e la. AWS SDK per Java AWS SDK per KotlinPer un elenco delle proprietà del sistema JVM supportate da, vedere il riferimento alle impostazioni. SDKs

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 tramite strumenti AWS SDKs e

Quando sviluppi un'applicazione AWS SDK o utilizzi AWS strumenti da utilizzare Servizi AWS, devi stabilire con che modo il codice o lo strumento si autentica. AWS Puoi configurare l'accesso programmatico alle AWS risorse in diversi modi, a seconda dell'ambiente in cui viene eseguito il codice e dell' AWS accesso a tua disposizione.

Le opzioni seguenti fanno parte della catena di fornitori di credenziali. Ciò significa che, configurando `credentials` i file condivisi AWS `config` e quelli condivisi di conseguenza, l' AWS SDK o lo strumento individueranno e utilizzeranno automaticamente quel metodo di autenticazione.

Scegli un metodo per autenticare il codice dell'applicazione

Scegli un metodo per autenticare le chiamate effettuate AWS dall'applicazione.

Stai eseguendo codice ALL'INTERNO di un Servizio AWS (come Amazon EC2, Lambda, Amazon ECS, Amazon EKS,)? CodeBuild

Se il codice continua a funzionare AWS, le credenziali possono essere rese automaticamente disponibili all'applicazione. Ad esempio, se l'applicazione è ospitata su Amazon Elastic Compute Cloud e a tale risorsa è associato un ruolo IAM, le credenziali vengono automaticamente rese disponibili all'applicazione. Allo stesso modo, se utilizzi contenitori Amazon ECS o Amazon EKS, le credenziali impostate per il ruolo IAM possono essere ottenute automaticamente dal codice in esecuzione all'interno del contenitore tramite la catena di provider di credenziali dell'SDK.

Il tuo codice si trova in un'istanza Amazon Elastic Compute Cloud?

Utilizzo dei ruoli IAM per autenticare le applicazioni distribuite su Amazon EC2— Usa i ruoli IAM per eseguire in modo sicuro la tua applicazione su un'istanza Amazon EC2.

Il tuo codice è contenuto in una AWS Lambda funzione?

Lambda crea un ruolo di esecuzione con autorizzazioni minime quando crei una funzione Lambda. L' AWS SDK o lo strumento utilizza quindi automaticamente il ruolo IAM collegato a Lambda in fase di esecuzione, tramite l'ambiente di esecuzione Lambda.

Il tuo codice è in Amazon Elastic Container Service (su Amazon EC2 o per AWS Fargate Amazon ECS)?

Usa IAM Role for Task. È necessario creare un ruolo di attività e specificare tale ruolo nella definizione dell'attività Amazon ECS. L' AWS SDK o lo strumento utilizza quindi automaticamente il ruolo IAM assegnato all'attività in fase di esecuzione, tramite i metadati Amazon ECS.

Il tuo codice è in Amazon Elastic Kubernetes Service?

Ti consigliamo di utilizzare Amazon EKS Pod Identities.

Nota: se ritieni che IAM roles for service accounts (IRSA) possa soddisfare meglio le tue esigenze specifiche, consulta la sezione Confronto tra EKS Pod Identity e IRSA nella Amazon EKS User Guide.

Il tuo codice è in esecuzione in AWS CodeBuild

Vedi Utilizzo di politiche basate sull'identità per. CodeBuild

Il tuo codice è in un altro? Servizio AWS

Consulta la guida dedicata per il tuo Servizio AWS. Quando esegui il codice AWS, la catena di fornitori di credenziali SDK può ottenere e aggiornare automaticamente le credenziali per te.

Stai creando applicazioni mobili o applicazioni web basate su client?

Se stai creando applicazioni mobili o applicazioni web basate su client che richiedono l'accesso a AWS, 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 quindi scambiarlo con credenziali di sicurezza temporanee in AWS quella mappa con un ruolo IAM con le autorizzazioni per utilizzare le risorse dell'utente. Account AWS

Per informazioni su come configurarlo per il tuo SDK o lo strumento, consulta. Assumere un ruolo con web identity o OpenID Connect per l'autenticazione e AWS SDKs gli strumenti

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.

Stai sviluppando ed eseguendo il codice LOCALMENTE?

ConsigliamoUtilizzo delle credenziali della console per l'autenticazione AWS SDKs e degli strumenti.

Dopo un rapido flusso di autenticazione basato su browser, genera AWS automaticamente credenziali temporanee che funzionano con strumenti di sviluppo locali come la CLI AWS e. AWS Strumenti per PowerShell AWS SDKs

Se utilizzi Identity Center per l'accesso all'account AWS

Usa IAM Identity Center per autenticare l' AWS SDK e gli strumenti se hai già accesso agli AWS account and/or necessari per gestire l'accesso per la tua forza lavoro. Come best practice di sicurezza, ti consigliamo di utilizzare AWS Organizations IAM Identity Center per gestire l'accesso a tutti i tuoi account. AWS 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 AWS IdP agli account. Per verificare se la tua regione supporta IAM Identity Center, consulta gli endpoint e le quote di Utilizzo di IAM Identity Center per autenticare AWS SDK e strumenti IAM Identity Center nell'Amazon Web Services General Reference.

Se stai cercando altri modi per autenticarti

Crea un utente IAM con i privilegi minimi con le autorizzazioni per accedere al tuo ruolo di destinazione. `sts:AssumeRole` Quindi configura il tuo profilo per assumere un ruolo utilizzando una `source_profile` configurazione per quell'utente.

Puoi anche utilizzare credenziali IAM temporanee tramite variabili di ambiente o il file di AWS credenziali condivise. Vedi Utilizzo di credenziali a breve termine per l'autenticazione e strumenti AWS SDKs .

Nota: solo in ambienti sandbox o di apprendimento, puoi prendere in considerazione l'utilizzo di credenziali a lungo termine per l'autenticazione e gli strumenti. AWS SDKs

Questo codice viene eseguito in locale o in una macchina virtuale ibrida/su richiesta (ad esempio un server che legge o scrive su Amazon S3 o Jenkins che distribuisce nel cloud)?

Stai utilizzando certificati client X.509?

Sì: vedi. Utilizzo di IAM Roles Anywhere per l'autenticazione AWS SDKs e gli strumenti 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'ambiente può connettersi in modo sicuro a un provider di identità federato (come Microsoft Entra o Okta) per richiedere credenziali temporanee? AWS

Sì: usa Provider di credenziali di processo

Utilizzato Provider di credenziali di processo per recuperare automaticamente le credenziali in fase di esecuzione. Questi sistemi potrebbero utilizzare uno strumento di supporto o un plug-in per ottenere le credenziali e potrebbero assumere un ruolo IAM dietro le quinte utilizzando. `sts:AssumeRole`

No: utilizza credenziali temporanee inserite tramite Gestione dei segreti AWS

Utilizza credenziali temporanee iniettate tramite. Gestione dei segreti AWSPer le opzioni per ottenere chiavi di accesso di breve durata, consulta Richiedere credenziali di sicurezza temporanee nella Guida per l'utente IAM. Per le opzioni sulla memorizzazione di queste credenziali temporanee, consulta. AWS chiavi di accesso

È possibile utilizzare queste credenziali per recuperare in modo sicuro autorizzazioni applicative più ampie da Secrets Manager, dove è possibile archiviare i segreti di produzione o le credenziali basate sui ruoli di lunga durata.

State utilizzando uno strumento di terze parti non presente? AWS

Utilizza la documentazione scritta dal tuo provider di terze parti per ottenere le migliori indicazioni su come ottenere le credenziali.

Se la tua terza parte non ha fornito la documentazione, puoi inserire credenziali temporanee in modo sicuro?

Sì: utilizza variabili di ambiente e credenziali temporanee. AWS STS

No: utilizza chiavi di accesso statiche archiviate in un gestore segreto crittografato (ultima risorsa).

Metodi di autenticazione

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

Se il codice continua a funzionare AWS, le credenziali possono essere rese automaticamente disponibili all'applicazione. Ad esempio, se l'applicazione è ospitata su Amazon Elastic Compute Cloud e a tale risorsa è associato un ruolo IAM, le credenziali vengono automaticamente rese disponibili all'applicazione. Allo stesso modo, se utilizzi contenitori Amazon ECS o Amazon EKS, le credenziali impostate per il ruolo IAM possono essere ottenute automaticamente dal codice in esecuzione all'interno del contenitore tramite la catena di provider di credenziali dell'SDK.

- Utilizzo dei ruoli IAM per autenticare le applicazioni distribuite su 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' AWS utilizzo di IAM Identity Center nei seguenti modi:
  - AWS CloudShellDa utilizzare per eseguire AWS CLI comandi dalla console.
  - 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 a AWS, 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 quindi scambiarlo con credenziali di sicurezza temporanee in AWS quella mappa con un ruolo IAM con le autorizzazioni per utilizzare le risorse dell'utente. Account AWS

Per informazioni su come configurarlo per il tuo SDK o lo strumento, consulta. Assumere un ruolo con web identity o OpenID Connect per l'autenticazione e AWS SDKs gli strumenti

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.

Metodi di autenticazione per il codice eseguito localmente (non in AWS)

- Utilizzo delle credenziali della console per l'autenticazione AWS SDKs e degli strumenti— Questa funzionalità funziona sia con l'interfaccia a riga di AWS comando che con gli strumenti PowerShell e fornisce credenziali aggiornabili che funzionano con strumenti di sviluppo locali come AWS CLI, Tools for and. PowerShell AWS
- Utilizzo di IAM Identity Center per autenticare AWS SDK e strumenti— Come best practice di sicurezza, consigliamo di utilizzare AWS Organizations IAM Identity Center per gestire l'accesso su tutti i tuoi. Account AWS Puoi creare utenti in AWS 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 Center endpoint e le quote nel. Riferimenti generali di Amazon Web Services
- Utilizzo di IAM Roles Anywhere per l'autenticazione AWS SDKs e gli strumenti— 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.
- Assumere un ruolo con AWS credenziali di autenticazione e strumenti AWS SDKs— Puoi assumere un ruolo IAM per accedere temporaneamente a AWS risorse a cui altrimenti non avresti accesso.
- Utilizzo delle chiavi di AWS accesso per l'autenticazione AWS SDKs e degli strumenti— Altre opzioni che potrebbero essere meno convenienti o che potrebbero aumentare il rischio di sicurezza per le AWS risorse.

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.

Plugin TIP (Trusted Identity Propagation) di IAM Identity Center a cui accedere Servizi AWS

- Utilizzo del plugin TIP per accedere Servizi AWS— Se stai creando un'applicazione per Amazon Q Business o un altro servizio che supporta la propagazione di identità affidabili e utilizzi il AWS SDK per Java o il AWS SDK per JavaScript, puoi utilizzare il plug-in TIP per un'esperienza di autorizzazione semplificata.

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 Q e Amazon CodeCatalyst.

- Accedi tramite la ID Builder AWS Guida per l'Accedi ad AWS utente: scopri come creare e utilizzare un ID Builder ID Builder AWS e scopri cosa offre.
- CodeCatalystconcetti - ID Builder AWS nella Amazon CodeCatalyst User Guide - Scopri come CodeCatalyst usa un ID Builder AWS.

Utilizzo delle credenziali della console per l'autenticazione AWS SDKs e degli strumenti

L'utilizzo delle credenziali della console è il metodo consigliato per fornire AWS le credenziali durante lo sviluppo di un' AWS applicazione nell'ambiente locale o in altri ambienti di servizio non di AWS elaborazione. Se stai sviluppando su una AWS risorsa, come Amazon Elastic Compute Cloud (Amazon EC2) AWS CloudShell oppure, ti consigliamo di ottenere le credenziali da quel servizio.

Puoi anche autenticarti tramite IAM Identity Center. Utilizzo di IAM Identity Center per autenticare AWS SDK e strumenti Questa opzione è un modo comune per le organizzazioni di gestire l'accesso per la propria forza lavoro e richiede l'attivazione di Identity Center.

Come funziona?

L'accesso per lo sviluppo AWS locale utilizzando le credenziali della console consente di utilizzare le credenziali di accesso esistenti AWS della Console di gestione per l'accesso programmatico ai servizi. AWS Dopo un flusso di autenticazione basato su browser, AWS genera credenziali temporanee che funzionano con strumenti di sviluppo locali come AWS CLI, Tools for and.

PowerShell AWS SDKs Questa funzionalità semplifica il processo di configurazione e gestione delle credenziali AWS CLI, soprattutto se si preferisce l'autenticazione interattiva alla gestione delle chiavi di accesso a lungo termine.

Con questo processo, puoi autenticarti utilizzando le credenziali root create durante la configurazione iniziale dell'account, gli utenti IAM o un'identità federata del tuo provider di identità.

Se lo utilizzi SDKs per lo sviluppo, i client SDK utilizzeranno le credenziali temporanee tramite. AWS SDKs e Tools: fornitori di credenziali standardizzati Puoi anche configurare il. Provider di credenziali di accesso

L'autenticazione tramite il comando login è supportata sia dalla AWS CLI che dagli strumenti per: PowerShell

- Accedi per lo sviluppo AWS locale utilizzando le credenziali della console
- Accedi utilizzando le credenziali della console nella guida per l' AWS Strumenti per PowerShell utente

Utilizzo di IAM Identity Center per autenticare AWS SDK e strumenti

AWS IAM Identity Center può essere utilizzato per fornire AWS credenziali durante lo sviluppo di un' AWS applicazione in ambienti di servizio non di AWS elaborazione. 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.

Utilizza l'autenticazione IAM Identity Center se utilizzi già Identity Center per l'accesso all' AWS account o devi gestire l'accesso per un'organizzazione.

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 SDK/tool autenticazione.

- 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.

Prerequisiti

Prima di iniziare questa procedura, dovresti aver completato quanto segue:

- Se non ne hai uno Account AWS, iscriviti a un Account AWS.
- Se non hai ancora abilitato IAM Identity Center, abilita IAM Identity Center seguendo le istruzioni nella Guida per l'AWS IAM Identity Center utente.

Configura l'accesso programmatico utilizzando IAM Identity Center

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

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

Non ho stabilito l’accesso tramite Centro identità IAM

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.

Al termine, è necessario disporre di 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 a livello di programmazione utilizzando il set di autorizzazioni.

Le implementazioni personalizzate possono dare luogo a esperienze diverse, ad esempio nomi di set di autorizzazioni diversi. In caso di dubbi su quale set di autorizzazioni utilizzare, contatta il team IT per 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 l'amministratore cloud ti ha concesso autorizzazioni `PowerUserAccess` (sviluppatore), potrai visualizzare gli Account AWS a 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 a livello di programmazione utilizzando il set di autorizzazioni.

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

Contatta il team IT per assistenza.

Fase 2: Configurazione SDKs e strumenti per utilizzare IAM Identity Center

1. Sulla tua macchina 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 `Issuer URL` e di cui hai bisogno per il passaggio successivo. Nota: `SSO Start URL` può essere usato in modo intercambiabile con. `Issuer URL`

3. Nel prompt dei AWS CLI comandi, esegui il comando. `aws configure sso` 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.

   a. Per visualizzare la richiesta`SSO Start URL`, immettete il valore ottenuto per. `Issuer URL`

   b. Per il nome del profilo CLI, ti consigliamo di inserire *`default`* 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 SDKs, 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. Comprendi la catena di fornitori di credenziali

Per un'analisi approfondita del modo in cui gli strumenti SDKs e utilizzano e aggiornano le credenziali utilizzando questa configurazione, consulta. Come viene risolta l'autenticazione AWS SDKs e gli strumenti di IAM Identity Center

Per configurare le impostazioni del provider IAM Identity Center direttamente nel `config` file condiviso, consulta questa Provider di credenziali IAM Identity Center guida.

Aggiornamento delle sessioni di accesso al portale

L'accesso alla fine scadrà e l'SDK o lo strumento riscontreranno un errore di autenticazione. Il momento in cui si verifica questa scadenza dipende dalla durata delle sessioni configurate. Per aggiornare nuovamente la sessione del portale di accesso quando necessario, utilizzare il comando AWS CLI per eseguire il `aws sso login` comando.

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

Come viene risolta l'autenticazione AWS SDKs e gli strumenti di IAM Identity Center

Termini pertinenti di IAM Identity Center

I seguenti termini ti aiutano a comprendere il processo e la configurazione alla base AWS IAM Identity Center. La documentazione per AWS SDK APIs 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
Centro identità	`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 IAM Identity Center rename nella Guida per l'utente di AWS IAM Identity Center .
Console IAM Identity Center Console amministrativa		La console che usi per configurare il Single Sign-On.
AWS accedere all'URL del portale		Un URL univoco per il tuo account IAM Identity Center, ad esempio`https://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 autorizzazioni		La sessione IAM che l'SDK utilizza internamente per

Nome IAM Identity Center	Nome dell'API SDK	Description
		effettuare le Servizio AWS chiamate. Nelle discussio ni informali, potresti vederla erroneamente definita «sessione di ruolo».
Credenziali del set di autorizza zioni	AWS credenziali 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 vedere che queste informazioni vengono 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 sigv4 APIs, 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

Avviare una sessione del portale di AWS accesso

- 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 CLI Inoltre 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 di. SDKs
- 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

SDKs fornisci l'accesso a Servizi AWS quando crei 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 sigv4 APIs 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' AWS accesso 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 Center utente.
- Tieni presente che le credenziali del set di autorizzazioni vengono anche chiamate credenziali e AWS credenziali 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 a Servizio 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.

Utilizzo di IAM Roles Anywhere per l'autenticazione AWS SDKs e gli strumenti

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 SDKs and the AWS CLI, puoi configurare le `credential_process` impostazioni nel file condiviso. AWS `config`

AWS CLI Supporta un provider di credenziali di processo che utilizza `credential_process` per l'autenticazione. SDKs 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. Console di gestione AWS 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

Assumere un ruolo con AWS credenziali di autenticazione e strumenti AWS SDKs

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 Service API Reference.

Per configurare l'SDK o lo strumento in modo che assuma 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 attendibile che utilizza il ruolo potrebbe essere una Servizio AWS o un'altra Account AWS. 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.

> **Note**
>
> È consigliabile utilizzare gli endpoint regionali ogni volta che è possibile e configurare i propri. AWS Regione AWS

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. In genere, questo tipo di assunzione di un ruolo viene utilizzato quando si dispone delle AWS credenziali per un account, ma l'applicazione richiede l'accesso alle risorse di un altro account.

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.

Utilizzalo `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 effettuano automaticamente la chiamata AWS STS `AssumeRole`API corrispondente. Per recuperare e utilizzare le 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 dei due `credential_source` o `source_profile`
- (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
credential_source = Ec2InstanceMetadata
```

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

[profile dev]
region = us-east-1
```

```
output = json
role_arn = arn:aws:iam::123456789012:role/my-role-name
source_profile = profile-with-user-that-can-assume-role
role_session_name = my_session
```

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.

Assumere un ruolo con web identity o OpenID Connect per l'autenticazione e AWS SDKs gli strumenti

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 Service API Reference.

Per configurare l'SDK o lo strumento in modo che assuma 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 provider di identità Web o una federazione OpenID Connect (OIDC) o SAML. Per ulteriori informazioni sui ruoli IAM, consulta Metodi per assumere un ruolo nella Guida per l'utente IAM.

Dopo aver configurato il ruolo IAM nel tuo SDK, se tale ruolo è configurato in modo da considerare attendibile il tuo provider di identità, puoi configurare ulteriormente il tuo SDK per assumere quel ruolo e ottenere credenziali temporanee AWS .

> **Note**
>
> È consigliabile utilizzare gli endpoint regionali ogni volta che è possibile e configurare i propri. AWS Regione AWS

Federazione con identità web o OpenID Connect

Puoi utilizzare i JSON Web Tokens (JWTs) di provider di identità pubblici, come Login With Amazon, Facebook, Google per ottenere AWS credenziali temporanee utilizzando.

`AssumeRoleWithWebIdentity` A seconda di come vengono utilizzati, JWTs possono essere chiamati token ID o token di accesso. È inoltre possibile utilizzare dati JWTs emessi da provider di identità (IdPs) compatibili con il protocollo di rilevamento di OIDC, come o. EntraId PingFederate

Se utilizzi Amazon Elastic Kubernetes Service, questa funzionalità offre la possibilità di specificare diversi ruoli IAM per ciascuno dei tuoi account di servizio in un cluster Amazon EKS. Questa funzionalità di Kubernetes viene distribuita JWTs ai tuoi pod, che vengono poi utilizzati da questo provider di credenziali per ottenere credenziali temporanee. AWS 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 JSON Web Token (JWT) per l'autenticazione, devi prima configurare l'identity provider (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 JSON Web Token (JWT) per l'autenticazione. AWS STS

Quando lo specifichi in un profilo, l'SDK o lo strumento effettua automaticamente la chiamata AWS STS `AssumeRoleWithWebIdentity`API corrispondente. 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 di SDKs and tools 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 SDKs, consulta Esempi di codice nella Amazon Cognito Developer Guide.

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.

Utilizzo delle chiavi di AWS accesso per l'autenticazione AWS SDKs e degli strumenti

L'uso delle chiavi di AWS accesso è un'opzione per l'autenticazione quando si utilizzano AWS SDKs strumenti.

Usa credenziali a breve termine

Ti consigliamo di configurare l'SDK o lo strumento da utilizzare per utilizzare Utilizzo di IAM Identity Center per autenticare AWS SDK e strumenti opzioni di durata estesa della sessione.

Tuttavia, per configurare direttamente l'SDK o le credenziali temporanee dello strumento, consulta. Utilizzo di credenziali a breve termine per l'autenticazione AWS SDKs e gli strumenti

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 AWS Utilizzando 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 Identità IAM (utenti, gruppi e ruoli) nella Guida per l'utente di IAM.

Per configurare un utente IAM per iniziare, consulta. Utilizzo di credenziali a lungo termine per l'autenticazione AWS SDKs e gli strumenti

Utilizzo di credenziali a breve termine per l'autenticazione AWS SDKs e gli strumenti

Ti consigliamo di configurare l' AWS SDK o lo strumento da utilizzare Utilizzo di IAM Identity Center per autenticare AWS SDK e strumenti 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.

Procedura consigliata: anziché gestire manualmente le chiavi di accesso e un token nel file delle credenziali, consigliamo all'applicazione di utilizzare credenziali temporanee fornite da:

- Un servizio di AWS elaborazione, ad esempio l'esecuzione della tua applicazione su Amazon Elastic Compute Cloud o in. AWS Lambda
- Un'altra opzione nella catena di fornitori di credenziali, ad esempio. Utilizzo di IAM Identity Center per autenticare AWS SDK e strumenti
- Oppure usa il Provider di credenziali di processo per recuperare le credenziali temporanee.

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 ha in genere 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.

Utilizzo di credenziali a lungo termine per l'autenticazione AWS SDKs e gli strumenti

> **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 includere file che contengono credenziali nell'area del progetto.
- Tieni presente che tutte le credenziali archiviate 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 Gestione dell'account AWS 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.

Fase 1: creare un utente IAM

- Crea il tuo utente IAM seguendo la procedura Creazione di utenti IAM (console) nella Guida per l'utente di IAM. Quando crei il tuo utente IAM:
  - Ti consigliamo di selezionare Fornisci l'accesso utente a Console di gestione AWS. 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 “Nozioni di base” utilizza il servizio Amazon S3 come esempio. Per fornire alla tua applicazione l’accesso completo ad Amazon S3, seleziona la policy `AmazonS3FullAccess` da allegare a questo utente.

- Puoi ignorare i passaggi facoltativi di tale procedura relativi all'impostazione dei limiti o dei tag di autorizzazione.

Fase 2: ottenere le chiavi di accesso

1. Nel riquadro di navigazione della console IAM, seleziona Utenti, quindi seleziona il **`User name`** dell'utente che hai creato in precedenza.
2. Nella pagina dell'utente, scegli la scheda 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 che con. AWS CLI SDKs
4. Per Crea chiave di accesso (fase 2), inserisci facoltativamente un tag e scegli Avanti.
5. Per Crea chiave di accesso (fase 3), seleziona Scarica il file.csv per salvare un file `.csv` con la chiave di accesso e la chiave di accesso segreta dell'utente IAM. Queste informazioni serviranno in seguito.

   > **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 file delle credenziali AWS `credentials` condiviso. Questo file è denominato `~/.aws/credentials` su sistemi Linux e macOS e `%USERPROFILE%\.aws\credentials` su Windows. Per ulteriori informazioni, consulta Ubicazione dei file delle credenziali.
2. Aggiungi il testo seguente al file `credentials` 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 autenticare le applicazioni distribuite su Amazon EC2

Questo esempio illustra la configurazione di un AWS Identity and Access Management ruolo con accesso Amazon S3 da utilizzare nella tua applicazione distribuita su un'istanza Amazon Elastic Compute Cloud.

Per eseguire la tua applicazione AWS SDK su un'istanza Amazon Elastic Compute Cloud, crea un ruolo IAM e consenti all' EC2 istanza Amazon di accedere a quel ruolo. Per ulteriori informazioni, consulta IAM Roles for Amazon EC2 nella Amazon EC2 User Guide.

Creazione di un ruolo IAM

L'applicazione AWS SDK che sviluppi probabilmente accede ad almeno una Servizio AWS per eseguire azioni. Crea un ruolo IAM che conceda le autorizzazioni necessarie per l'esecuzione dell'applicazione.

Questa procedura crea un ruolo che garantisce l'accesso in sola lettura ad Amazon S3, ad esempio. Molte guide AWS SDK contengono tutorial introduttivi tratti da Amazon S3.

1. Accedi Console di gestione AWS 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 creerai la tua EC2 istanza Amazon.

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

Puoi creare e avviare un' EC2 istanza Amazon utilizzando il tuo ruolo IAM effettuando le seguenti operazioni:

- Segui Quickly launch an instance nella Amazon EC2 User Guide. Tuttavia, prima della fase finale di invio, procedi anche come segue:
  - In Dettagli avanzati, per il profilo IAM Instance, scegli il ruolo che hai creato nel passaggio precedente.

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

Connect all' EC2 istanza

Connettiti all' EC2 istanza Amazon in modo da poter trasferire l'applicazione su di essa e quindi eseguire l'applicazione. Avrai bisogno del file che contiene la parte privata della coppia di chiavi che hai usato in Key pair (login) quando hai creato l'istanza, ovvero il file PEM.

Puoi farlo seguendo le indicazioni per il tuo tipo di istanza: Connetti alla tua istanza Linux o Connect alla tua istanza Windows. Quando ti connetti, fallo in modo da poter trasferire i file dalla macchina di sviluppo all'istanza.

> **Note**
>
> Sul terminale Linux o macOS, puoi usare il comando secure copy per copiare la tua applicazione. Da usare `scp` con una key pair, puoi usare il seguente comando:`scp -i path/to/key file/to/copy ec2-user@ec2-xx-xx-xxx-xxx.compute.amazonaws.com:~`.
> Per ulteriori informazioni per Windows, consulta Trasferimento di file su istanze Windows.

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 sull'istanza EC2

1. Copia i file dell'applicazione dall'unità locale all' EC2 istanza Amazon.

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 a Console di gestione AWS e apri la EC2 console Amazon all'indirizzo https://console.aws.amazon.com/ec2/.
    b. Selezionare l'istanza.
    c. Scegli Azioni, Sicurezza, quindi scegli Modifica ruolo IAM.
    d. Per il ruolo IAM, scollega il ruolo IAM scegliendo Nessun ruolo IAM.
    e. Scegli Aggiorna ruolo IAM.
    f. Esegui nuovamente l'applicazione e conferma che restituisca un errore di autorizzazione.

Utilizzo del plugin TIP per accedere Servizi AWS

La propagazione affidabile dell'identità (TIP) è una funzionalità AWS IAM Identity Center che consente agli amministratori di concedere autorizzazioni Servizi AWS in base agli attributi degli utenti, come le associazioni di gruppo. Con la propagazione affidabile delle identità, il contesto dell'identità viene aggiunto a un ruolo IAM per identificare l'utente che richiede l'accesso alle risorse. AWS Questo contesto viene propagato ad altri. Servizi AWS

Il contesto di identità comprende le informazioni che vengono Servizi AWS utilizzate per prendere decisioni di autorizzazione quando ricevono richieste di accesso. Queste informazioni includono metadati che identificano il richiedente (ad esempio, un utente IAM Identity Center), il Servizio AWS cui accesso è richiesto (ad esempio, Amazon Redshift) e l'ambito di accesso (ad esempio, l'accesso in sola lettura). La ricezione Servizio AWS utilizza questo contesto e tutte le autorizzazioni assegnate all'utente per autorizzare l'accesso alle sue risorse. Per ulteriori informazioni, vedere la panoramica sulla propagazione delle identità affidabili nella Guida per l' AWS IAM Identity Center utente.

Il plug-in TIP può essere utilizzato con dispositivi Servizi AWS che supportano la propagazione di identità affidabili. Come caso d'uso di riferimento, consulta Configurazione di un'applicazione Amazon Q Business utilizzando AWS IAM Identity Center nella Amazon Q Business User Guide.

> **Note**
>
> Se utilizzi Amazon Q Business, consulta Configurazione di un'applicazione Amazon Q Business AWS IAM Identity Center per istruzioni specifiche sul servizio.

Prerequisiti per l'utilizzo del plug-in TIP

Per il funzionamento del plugin sono necessarie le seguenti risorse:

1. È necessario utilizzare il AWS SDK per Java o il AWS SDK per JavaScript.
2. Verifica che il servizio che stai utilizzando supporti la propagazione dell'identità affidabile.

   Consulta la colonna Abilita la propagazione dell'identità affidabile tramite IAM Identity Center delle applicazioni AWS gestite che si integrano con IAM Identity Center nella Guida per l'AWS IAM Identity Center utente.
3. Abilita IAM Identity Center e la propagazione affidabile delle identità.

   Consulta i prerequisiti e le considerazioni del TIP nella Guida per l'AWS IAM Identity Center utente.
4. È necessario disporre di un' Identity-Center-integratedapplicazione.

   Vedi le applicazioni AWS gestite o le applicazioni gestite dal cliente nella Guida AWS IAM Identity Center per l'utente.
5. È necessario configurare un Trusted Token Issuer (TTI) e connettere il servizio a IAM Identity Center.

   Consulta Prerequisiti per emittenti di token affidabili e Attività per la configurazione di un emittente di token affidabile nella Guida per l'utente.AWS IAM Identity Center

Per utilizzare il plug-in TIP nel codice

1. Crea un'istanza del plugin affidabile per la propagazione delle identità.
2. Crea un'istanza del client di servizio per interagire con il tuo Servizio AWS e personalizza il client di servizio aggiungendo il plug-in di propagazione dell'identità affidabile.

Il plugin TIP accetta i seguenti parametri di input:

- **`webTokenProvider`**: una funzione che il cliente implementa per ottenere un token OpenID dal proprio provider di identità esterno.
- **`accessRoleArn`**: l'ARN del ruolo IAM che deve essere assunto dal plug-in con il contesto di identità dell'utente per ottenere le credenziali con identità avanzata.
- **`applicationArn`**: La stringa identificativa univoca per il client o l'applicazione. Questo valore è un ARN dell'applicazione con OAuth sovvenzioni configurate.

- **`ssoOidcClient`**: (Facoltativo) Un client SSO OIDC, ad esempio `SsoOidcClient`per Java o `client-sso-oidc`for JavaScript, con configurazioni definite dal cliente. Se non viene fornito, verrà istanziato e utilizzato un client OIDC che utilizza `applicationRoleArn`.
- **`stsClient`**: (Facoltativo) Un AWS STS client con configurazioni definite dal cliente, utilizzato per assumere il contesto di identità dell'utente. `accessRoleArn` Se non viene fornito, `applicationRoleArn` verrà istanziato e utilizzato un AWS STS client che utilizza.
- **`applicationRoleArn`**: (Facoltativo) L'ARN del ruolo IAM da assumere in `AssumeRoleWithWebIdentity` modo che l'OIDC e AWS STS i client possano essere avviati.
  - Se non viene fornito, devono essere forniti entrambi i parametri `ssoOidcClient` and`stsClient`.
  - Se fornito, non `applicationRoleArn` può avere lo stesso valore del `accessRoleArn` parametro. `applicationRoleArn`viene utilizzato per creare STSClient, che viene utilizzato per assumere AccessRole. Se si utilizza lo stesso ruolo per entrambi `applicationRole` e`accessRole`, ciò significherebbe utilizzare un ruolo per assumere se stesso (presupposto del ruolo autonomo), cosa sconsigliata da. AWS Vedi l'annuncio per maggiori dettagli.

Considerazioni relative **`ssoOidcClient`** a **`stsClient`** e parametri **`applicationRoleArn`**

Quando configurate il plugin TIP, tenete conto dei seguenti requisiti di autorizzazione in base ai parametri forniti:

- Se stai fornendo `ssoOidcClient` e`stsClient`:
  - Le credenziali sul centro di identità `ssoOidcClient` devono essere `oauth:CreateTokenWithIAM` autorizzate a chiamare il centro di identità per ottenere il contesto utente specifico del centro di identità.
  - Le credenziali `stsClient` devono essere attivate `sts:AssumeRole` e le `sts:SetContext` autorizzazioni devono essere attivate. `accessRole` `accessRole`deve inoltre essere configurato con una relazione di fiducia con le credenziali attivate. `stsClient`
- Se stai fornendo`applicationRoleArn`:
  - `applicationRole`dovrebbe avere le `oauth:CreateTokenWithIAM` autorizzazioni `sts:AssumeRole` e `sts:SetContext` le autorizzazioni sulle risorse richieste (istanza IdC`accessRole`), poiché verrà utilizzata per creare client OIDC e STS.

- `applicationRole`dovrebbe avere una relazione di fiducia con il provider di identità utilizzato per generare il`webToken`, poiché `webToken` verrà utilizzato per assumere l'ApplicationRole tramite la chiamata del plug-in. AssumeRoleWithWebIdentity

Configurazione di esempio ApplicationRole :

Politica di fiducia con provider di token Web:

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

Politica di autorizzazione:

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

```
        {
            "Effect": "Allow",
            "Action": [
                "sso-oauth:CreateTokenWithIAM"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

Esempi di codice che utilizzano TIP

Gli esempi seguenti mostrano come implementare il plug-in TIP nel codice utilizzando AWS SDK per Java o il AWS SDK per JavaScript.

Java

Per utilizzare il plugin TIP nel tuo AWS SDK per Java progetto, devi dichiararlo come dipendenza nel file del `pom.xml` tuo progetto.

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

Nel codice sorgente, includete la dichiarazione del pacchetto richiesta per. `software.amazon.awssdk.trustedidentitypropagation`

Gli esempi seguenti mostrano due modi per creare un'istanza del plugin di propagazione dell'identità affidabile e aggiungerla a un client di servizio. Entrambi gli esempi utilizzano Amazon S3 come servizio e lo utilizzano `S3AccessGrantsPlugin` per gestire le autorizzazioni specifiche degli utenti, ma possono essere applicati a qualsiasi sistema Servizio AWS che supporti la propagazione dell'identità affidabile (TIP).

> **Note**
>
> Per questi esempi, è necessario configurare le autorizzazioni specifiche dell'utente da S3 Access Grants. Consulta la documentazione di S3 Access Grants per maggiori dettagli.

Opzione 1: crea e trasferisci client OIDC e STS

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

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

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

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

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

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

Opzione 2: Passa applicationRoleArn e rimanda la creazione del client al plugin

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

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

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

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

Per ulteriori dettagli e fonti, vedi trusted-identity-propagation-javasu GitHub.

JavaScript

Esegui il seguente comando per installare il pacchetto del plug-in di autenticazione TIP nel tuo AWS SDK per JavaScript progetto:

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

Il finale `package.json` dovrebbe includere una dipendenza simile alla seguente:

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

Nel codice sorgente, importa la `TrustedIdentityPropagationExtension` dipendenza richiesta.

Gli esempi seguenti mostrano due modi per creare un'istanza del plugin di propagazione dell'identità affidabile e aggiungerla a un client di servizio. Entrambi gli esempi utilizzano Amazon S3 come servizio e Amazon S3 Access Grants per gestire le autorizzazioni specifiche degli utenti, ma possono essere applicati a qualsiasi sistema Servizio AWS che supporti la propagazione dell'identità affidabile (TIP).

> **Note**
>
> Per questi esempi, è necessario configurare le autorizzazioni specifiche dell'utente da Amazon S3 Access Grants, consulta la documentazione di Amazon S3 Access Grants per maggiori dettagli.

Opzione 1: crea e trasferisci client OIDC e STS

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

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

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

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

  const credentials = response.Credentials;

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

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

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

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

Opzione 2: Passa applicationRoleArn e rimanda la creazione del client al plugin

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

const s3ControlClient = new S3ControlClient({
    region: "us-east-1",
    extensions: [
```

```
        TrustedIdentityPropagationExtension.create({
            webTokenProvider: async () => {
                return 'ID_TOKEN_FROM_YOUR_IDENTITY_PROVIDER';
            },
            accessRoleArn: accessRoleArn,
            applicationRoleArn: applicationRoleArn,
            applicationArn: applicationArn,
        }),
    ],
});

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

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

  const credentials = response.Credentials;

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

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

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

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

Per ulteriori dettagli e fonti, vedi trusted-identity-propagation-jssu GitHub.

AWS SDKs riferimento alle impostazioni e agli strumenti

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

È possibile trovare informazioni dettagliate sulle impostazioni di configurazione nelle seguenti sezioni:

- AWS SDKs e Tools: fornitori di credenziali standardizzati— Fornitori di credenziali comuni standardizzati su più piattaforme. SDKs
- AWS SDKs e funzionalità standardizzate di Tools— Funzionalità comuni standardizzate su più piattaforme. SDKs

Creazione di clienti di servizio

Per accedere a livello di codice Servizi AWS, SDKs utilizza un client class/object per ciascuno. Servizio AWS Ad esempio, se la tua applicazione deve accedere ad Amazon EC2, crea un oggetto EC2 client Amazon per interfacciarsi con quel servizio. Quindi utilizzi il client del servizio per effettuare richieste in merito Servizio AWS. Nella maggior parte dei casi SDKs, 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 che sono supportate dalla maggior parte SDKs e hanno un ampio impatto su tutti. Servizi AWS Tutte SDKs hanno una serie di luoghi (o fonti) che controllano per trovare un valore per le impostazioni globali. Quanto segue è l'impostazione della priorità 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 Strumenti per PowerShell, queste assumono la forma di parametri specifici per operazione

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 controllata. 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 SDKs strumenti potrebbero eseguire il controllo in un ordine diverso. Inoltre, alcuni SDKs strumenti supportano altri metodi di memorizzazione e recupero dei parametri. Ad esempio, AWS SDK per .NET supporta una fonte aggiuntiva denominata 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.

Informazioni sulle pagine delle impostazioni di questa guida

Le pagine all'interno della sezione di riferimento alle Impostazioni di questa guida descrivono in dettaglio le impostazioni disponibili che possono essere configurate tramite vari meccanismi. Le

tabelle che seguono elencano le impostazioni dei file di configurazione e credenziali, le variabili di ambiente e (per Java e Kotlin SDKs) le impostazioni JVM che possono essere utilizzate al di fuori del codice per configurare la funzionalità. Ogni argomento collegato in ogni elenco porta alla pagina delle impostazioni corrispondente.

- `Config`elenco delle impostazioni dei file
- `Credentials`elenco delle impostazioni dei file
- elenco delle variabili di ambiente
- Elenco delle proprietà del sistema JVM

Ogni provider di credenziali o funzionalità dispone di una pagina in cui sono elencate le impostazioni utilizzate per configurare tale funzionalità. Per ogni impostazione, è spesso possibile impostare il valore aggiungendo l'impostazione a un file di configurazione o impostando una variabile di ambiente oppure (solo per Java e Kotlin) impostando una proprietà di sistema JVM. Ogni impostazione elenca tutti i metodi supportati per impostare il valore in un blocco sopra i dettagli della descrizione. Sebbene la precedenza vari, la funzionalità risultante è la stessa indipendentemente da come viene impostata.

La descrizione includerà l'eventuale valore predefinito che avrà effetto se non si esegue alcuna operazione. Definisce inoltre il valore valido per quell'impostazione.

Ad esempio, diamo un'occhiata a un'impostazione dalla pagina delle Richiesta di compressione funzionalità.

Le informazioni dell'impostazione di `disable_request_compression` esempio documentano quanto segue:

- Esistono tre modi equivalenti per controllare la compressione delle richieste al di fuori del codebase. Puoi eseguire una delle seguenti operazioni:
  - Impostalo nel tuo file di configurazione usando `disable_request_compression`
  - Impostala come variabile di ambiente usando `AWS_DISABLE_REQUEST_COMPRESSION`
  - Oppure, se stai usando Java o Kotlin SDK, impostalo come proprietà di sistema JVM usando `aws.disableRequestCompression`

> **Note**
>
> Potrebbe esserci anche un modo per configurare la stessa funzionalità direttamente nel codice, ma questo riferimento non copre questo aspetto poiché è unico per ogni SDK.

Se desideri impostare la configurazione nel codice stesso, consulta la guida SDK o il riferimento API specifico.

- Se non fai nulla, il valore predefinito sarà. `false`
- Gli unici valori validi per questa impostazione booleana sono `true` e. `false`

Nella parte inferiore di ogni pagina di funzionalità è presente una tabella Support by AWS SDKs e strumenti.

Questa tabella mostra se l'SDK supporta le impostazioni elencate nella pagina. La `Supported` colonna indica il livello di supporto con i seguenti valori:

- `Yes`— Le impostazioni sono completamente supportate dall'SDK così come scritto.
- `Partial`— Alcune impostazioni sono supportate o il comportamento si discosta dalla descrizione. Infatti`Partial`, una nota aggiuntiva indica la deviazione.
- No— Nessuna delle impostazioni è supportata. Ciò non afferma se sia possibile ottenere la stessa funzionalità nel codice; indica solo che le impostazioni di configurazione esterne elencate non sono supportate.

`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. SDKs 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	
`account_id_endpoint_mode`	Endpoint basati sull'account	
`api_versions`	Impostazioni generali di configurazione	

Nome dell'impostazione	Informazioni
`auth_scheme_preference`	Schema di autenticazione
`aws_access_key_id`	AWS chiavi di accesso
`aws_account_id`	endpoint basati sull'account
`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_host_prefix_injection`	Iniezione del prefisso dell'host
`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 EC2 istanze Amazon
`metadata_service_timeout`	Metadati delle EC2 istanze Amazon
`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_checksum_calculation`	Protezioni per l'integrità dei dati per Amazon S3
`request_min_compression_size_bytes`	Richiedi la compressione
`response_checksum_validation`	Protezioni per l'integrità dei dati per Amazon S3
`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_express_session_auth`	Autenticazione della sessione S3 Express One Zone
`s3_disable_multiregion_access_points`	Punti di accesso multi-Regione di Amazon S3

Nome dell'impostazione	Informazioni
s3_use_arn_region	Access point Amazon S3
sdk_ua_app_id	ID dell'applicazione
sigv4a_signing_region_set	Schema di autenticazione
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
sso_role_name	Provider di credenziali IAM Identity Center
sso_start_url	Provider di credenziali IAM Identity Center
sts_regional_endpoints	AWS STS Endpoint regionali
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. SDKs 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 SDKs sono elencate nella tabella seguente. Sono globali e riguardano tutti Servizi AWS. SDKs 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_ACCOUNT_ID`	endpoint basati sull'account	

Nome dell'impostazione	Informazioni	
AWS_ACCOUNT_ID_ENDPOINT_MODE	Endpoint basati sull'account	
AWS_AUTH_SCHEME_PREFERENCE	Schema di autenticazione	
AWS_CA_BUNDLE	Impostazioni generali di configurazione	
AWS_CONFIG_FILE	Individuazione e modifica della posizione dei `credentials` file condivisi `config`AWS SDKs e degli strumenti	
AWS_CONTAINER_AUTHORIZATION_TOKEN	Fornitore di credenziali per container	
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 intelligente	
AWS_DISABLE_HOST_PREFIX_INJECTION	Iniezione del prefisso dell'host	
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	

Nome dell'impostazione	Informazioni	
`AWS_ENDPOINT_URL`	Endpoint specifici del servizio	
`AWS_ENDPOINT_URL_<SERVICE>`	Endpoint specifici del servizio	
`AWS_IGNORE_CONFIGURED_ENDPOINT_URLS`	Endpoint specifici del servizio	
`AWS_MAX_ATTEMPTS`	Comportamento dei nuovi tentativi	
`AWS_METADATA_SERVICE_NUM_ATTEMPTS`	Metadati delle EC2 istanze Amazon	
`AWS_METADATA_SERVICE_TIMEOUT`	Metadati delle EC2 istanze Amazon	
`AWS_PROFILE`	Utilizzo di strumenti `config` e AWS SDKs configurazioni condivisi e di `credentials` file a livello globale	
`AWS_REGION`	Regione AWS	
`AWS_REQUEST_CHECKSUM_CALCULATION`	Protezioni per l'integrità dei dati per Amazon S3	

Nome dell'impostazione	Informazioni	
AWS_REQUEST_MIN_COMPRESSION_SIZE_BYTES	Richiedi la compressione	
AWS_RESPONSE_CHECKSUM_VALIDATION	Protezioni per l'integrità dei dati per Amazon S3	
AWS_RETRY_MODE	Comportamento dei nuovi tentativi	
AWS_ROLE_ARN	Assumi il ruolo di fornitore di credenziali	
AWS_ROLE_SESSION_NAME	Assumi il ruolo di fornitore di credenziali	
AWS_S3_DISABLE_EXPRESS_SESSION_AUTH	Autenticazione della sessione S3 Express One Zone	
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	

Nome dell'impostazione	Informazioni	
AWS_SESSION_TOKEN	AWS chiavi di accesso	
AWS_SHARED_CREDENTIALS_FILE	Individuazione e modifica della posizione dei `credentials` file condivisi `config`AWS SDKs e degli strumenti	
AWS_SIGV4A_SIGNING_REGION_SET	Schema di autenticazione	
AWS_STS_REGIONAL_ENDPOINTS	AWS STS Endpoint regionali	
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 per Java e AWS SDK per 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.accountId`	endpoint basati sull'account
`aws.accountIdEndpointMode`	Endpoint basati sull'account
`aws.authSchemePreference`	Schema di autenticazione
`aws.configFile`	Individuazione e modifica della posizione dei `credentials` file condivisi `config`AWS SDKs e degli strumenti
`aws.defaultsMode`	Impostazioni predefinite di configurazione intelligenti
`aws.disableEc2MetadataV1`	Fornitore di credenziali IMDS
`aws.disableHostPrefixInjection`	Iniezione del prefisso dell'host
`aws.disableRequestCompression`	Richiedi la compressione
`aws.disableS3ExpressAuth`	Autenticazione della sessione S3 Express One Zone

Nome dell'impostazione	Informazioni
`aws.ec2MetadataServiceEndpoint`	Fornitore di credenziali IMDS
`aws.ec2MetadataServiceEndpointMode`	Fornitore di credenziali IMDS
`aws.endpointDiscoveryEnabled`	Individuazione degli endpoint
`aws.endpointUrl`	Endpoint specifici del servizio
`aws.endpointUrl<ServiceName>`	Endpoint specifici del servizio
`aws.ignoreConfiguredEndpointUrls`	Endpoint specifici del servizio
`aws.maxAttempts`	Comportamento dei nuovi tentativi
`aws.profile`	Utilizzo di `credentials` file `config` e file condivisi per configurare AWS SDKs e utilizzare strumenti a livello globale
`aws.region`	Regione AWS
`aws.requestChecksumCalculation`	Protezioni per l'integrità dei dati per Amazon S3

Nome dell'impostazione	Informazioni
`aws.requestMinCompressionSizeBytes`	Richiedi la compressione
`aws.responseChecksumValidation`	Protezioni per l'integrità dei dati per Amazon S3
`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
`aws.sessionToken`	AWS chiavi di accesso
`aws.sharedCredentialsFile`	Individuazione e modifica della posizione dei `credentials` file condivisi `config`AWS SDKs e degli strumenti

Nome dell'impostazione	Informazioni	
`aws.useDualstackEndpoint`	Endpoint dual-stack e FIPS	
`aws.useFipsEndpoint`	Endpoint dual-stack e FIPS	
`aws.webIdentityTokenFile`	Assumi il ruolo di fornitore di credenziali	
`sdk.ua.appId`	ID dell'applicazione	

AWS SDKs e Tools: fornitori di credenziali standardizzati

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

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

Argomenti

- Comprendi la catena di fornitori di credenziali
- Catene di fornitori di credenziali specifiche per SDK e strumenti
- AWS chiavi di accesso
- Provider di credenziali 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

Comprendi la catena di fornitori di credenziali

Tutti SDKs hanno una serie di luoghi (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.

Quando si utilizza uno dei provider di credenziali standardizzati, cercano AWS SDKs sempre di rinnovare automaticamente le credenziali quando scadono. La catena di fornitori di credenziali integrata offre all'applicazione la possibilità di aggiornare le credenziali indipendentemente dal provider utilizzato nella catena. Per eseguire questa operazione non è necessario alcun codice aggiuntivo per l'SDK.

Sebbene la catena distinta utilizzata da ciascun SDK vari, la maggior parte delle volte include fonti come le seguenti:

Fornitore di credenziali	Description
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 JSON Web Token (JWT) da (). AWS Security Token Service AWS STS
Provider di credenziali di accesso	Ottieni le credenziali per una sessione di console nuova o esistente a cui hai effettuato l'accesso.
Provider di credenziali IAM Identity Center	Ottieni credenziali da. AWS IAM Identity Center

Fornitore di credenziali	Description
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	Provider 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 distribuite tramite il servizio di metadati di 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 ilUtilizzo di `credentials` file `config` e file condivisi per configurare AWS SDKs e utilizzare strumenti a livello globale. Per ulteriori informazioni, consulta Precedenza delle impostazioni.

Catene di fornitori di credenziali specifiche per SDK e strumenti

Per accedere direttamente ai dettagli della catena di fornitori di credenziali specifica del tuo SDK o dello strumento, scegli il tuo SDK o lo strumento tra i seguenti:

- AWS CLI
- SDK per C++
- SDK for Go
- SDK per Java

- SDK per JavaScript
- SDK per Kotlin
- SDK per.NET
- SDK per PHP
- SDK per Python (Boto3)
- SDK per Ruby
- SDK per Rust
- SDK per Swift
- Strumenti per PowerShell

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, consultaUtilizzo di credenziali a breve termine per l'autenticazione AWS SDKs e gli strumenti.

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
```

Support by AWS SDKs and tools

Di seguito sono SDKs supportate le funzionalità e le impostazioni descritte in questo argomento. Vengono annotate eventuali eccezioni parziali. Qualsiasi impostazione delle proprietà del sistema JVM è supportata solo da AWS SDK per Java and the. AWS SDK per Kotlin

SDK	Sı a	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	Si a	Note o ulteriori informazioni
SDK per.NET 4.x	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 per Swift	Sì	
Strumenti per V5 PowerShell	Sì	
Strumenti per PowerShell V4	Sì	Variabili di ambiente non supportate.

Provider di credenziali di accesso

È possibile utilizzare le credenziali di accesso esistenti AWS della Console di gestione per acquisire credenziali a breve termine da utilizzare per l'accesso programmatico. Dopo aver completato il flusso di autenticazione basato su browser, AWS genera credenziali temporanee che funzionano con strumenti di sviluppo locali come AWS CLI, Tools for and. AWS PowerShell AWS SDKs

Per generare queste credenziali, eseguire il `aws login` comando nella AWS CLI o `Invoke-AWSLogin` il cmdlet AWS in Tools for. PowerShell Le credenziali a breve termine risultanti verranno memorizzate nella cache locale, dove potranno essere riutilizzate da. AWS SDKs Le credenziali a breve termine scadono dopo 15 minuti, ma l'interfaccia CLI e le SDKs aggiorneranno automaticamente secondo necessità per un massimo di 12 ore. Alla scadenza del token di aggiornamento, ti verrà richiesto di accedere nuovamente tramite la CLI o. PowerShell

Il comando login aggiornerà il profilo specificato con l'`login_session`impostazione, che memorizza l'identità della sessione della console di gestione selezionata durante il flusso di lavoro di accesso.

```
[profile console]
login_session = arn:aws:iam::0123456789012:user/username
```

```
region = us-west-2
```

Per impostazione predefinita, le credenziali a breve termine e il token di aggiornamento vengono archiviati in un file JSON nella `~/.aws/login/cache` directory su Linux e macOS o su Windows. `%USERPROFILE%\.aws\login\cache` Il nome del file si basa sul nome della sessione di accesso. È possibile sovrascrivere la directory impostando la `AWS_LOGIN_CACHE_DIRECTORY` variabile di ambiente.

Impostazioni del provider di accesso

Configura questa funzionalità utilizzando quanto segue:

`AWS_LOGIN_CACHE_DIRECTORY`- variabile di ambiente

Directory alternativa in cui la CLI e la CLI SDKs memorizzeranno le credenziali memorizzate nella cache che mappano a un profilo di sessione di accesso.

Valore predefinito: `~/.aws/login/cache` su Linux e macOS o `%USERPROFILE%\.aws\login\cache` su Windows.

Support by AWS SDKs and tools

Di seguito sono SDKs supportate 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 per Java and the. AWS SDK per Kotlin

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

SDK	Si a	Note o ulteriori informazioni
SDK per 3.x JavaScript	Sì	
SDK per 2.x JavaScript	No	
SDK per Kotlin	Sì	
SDK per.NET 4.x	Sì	
SDK per.NET 3.x	Sì	
SDK per PHP 3.x	Sì	
SDK per Python (Boto3)	Sì	Richiede CRT
SDK per Ruby 3.x	Sì	
SDK per Rust	Sì	
Strumenti per PowerShell V5	Sì	
Strumenti per PowerShell V4	No	

Assumi il ruolo di fornitore di credenziali

> **Note**
>
> Per informazioni sulla comprensione del layout delle pagine delle impostazioni o sull'interpretazione della tabella Support by AWS SDKs and tools riportata di seguito, vedereInformazioni sulle pagine delle impostazioni di questa guida.

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 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, Assumere un ruolo con AWS credenziali di autenticazione e strumenti AWS SDKs 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: specifica che l'SDK o lo strumento deve utilizzare il ruolo IAM collegato al contenitore Amazon ECS o il ruolo IAM collegato al contenitore Amazon EKS 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 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. Per ulteriori informazioni sulla MFA, consulta Autenticazione a AWS più fattori in IAM nella IAM User Guide.

Valore predefinito: Nessuno.

Valori validi: il valore può essere un numero di serie per un dispositivo hardware (ad esempioGAHT12345678) o un Amazon Resource Name (ARN) per un dispositivo MFA virtuale. Il formato dell'ARN è: `arn:aws:iam::account-id:mfa/mfa-device-name`

Esempio di impostazione in un `config` file:

Questo esempio presuppone un dispositivo MFA virtuale, `MyMFADevice` chiamato, che sia stato creato per l'account e abilitato per un utente.

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

`role_arn`- impostazione condivisa AWS **`config`** dei file, **`AWS_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_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 viene visualizzato nei AWS CloudTrail registri delle voci associate a questa sessione, il che può essere utile durante il controllo. Per i dettagli, consulta l'elemento CloudTrail UserIdentity nella Guida per l'AWS CloudTrail utente.

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
role_session_name = ProfileARoleSession
```

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

Nell'esempio precedente, il `A` profilo indica all'SDK o allo strumento di cercare automaticamente le credenziali per il profilo collegato. `B` In questo caso, il `B` profilo utilizza lo strumento di supporto alle credenziali fornito da per ottenere le credenziali per Utilizzo di IAM Roles Anywhere per l'autenticazione AWS SDKs e gli strumenti l'SDK. AWS 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. Ogni azione eseguita dal profilo `A` ha il nome della sessione del ruolo incluso nei CloudTrail log.

Per un secondo esempio di concatenamento dei ruoli, è possibile utilizzare la seguente configurazione se hai un'applicazione su un'istanza Amazon Elastic Compute Cloud e desideri che tale applicazione assuma un altro ruolo.

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

[profile B]
credential_source=Ec2InstanceMetadata
```

Il profilo `A` utilizzerà le credenziali dell'istanza Amazon EC2 per assumere il ruolo specificato e rinnoverà le credenziali automaticamente.

`web_identity_token_file`- impostazione di file condivisi AWS **`config`**, **`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.

Support by AWS SDKs and tools

Di seguito sono SDKs supportate 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 per Java and the. AWS SDK per Kotlin

SDK	Supporta	Note o ulteriori informazioni
AWS CLI v2	Sì	
SDK per C++	Parziale	`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	Parziale	`mfa_serial` non supportato. `duration_seconds` non supportato.
SDK per Java 1.x	Parziale	`credential_source` non supportato. `mfa_serial` non supportato. Le proprietà del sistema JVM non sono supportate.
SDK per 3.x JavaScript	Sì	
SDK per 2.x JavaScript	Parziale	`credential_source` non supportato.
SDK per Kotlin	Sì	

SDK	Su a	Note o ulteriori informazioni
SDK per.NET 4.x	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 per Swift	Sì	
Strumenti per V5 PowerShell	Sì	
Strumenti per PowerShell V4	Sì	

Provider di credenziali per container

> **Note**
>
> Per informazioni sulla comprensione del layout delle pagine delle impostazioni o sull'interpretazione della tabella Support by AWS SDKs and tools riportata di seguito, vedereInformazioni sulle pagine delle impostazioni di questa guida.

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). SDKs tenta 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 strumenti

SDKs 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 strumenti SDKs 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. Sono inclusi 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
```

or

```
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
```

Support by AWS SDKs and tools

Di seguito sono SDKs supportate 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 per Java and the. AWS SDK per Kotlin

SDK	Su a	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	Quando Lambda SnapStart è attivata `AWS_CONTAINER_CREDENTIALS_FULL_URI` e `AWS_CONTAINER_AUTHORIZATION_TOKEN` viene utilizzata automaticamente per l'autenticazione.
SDK per Java 1.x	Sì	Quando Lambda SnapStart è attivata `AWS_CONTAINER_CREDENTIALS_FULL_URI` e `AWS_CONTAINER_AUTHORIZATION_TOKEN` viene utilizzata automaticamente per l'autenticazione.
SDK per 3.x JavaScript	Sì	
SDK per 2.x JavaScript	Sì	
SDK per Kotlin	Sì	
SDK per.NET 4.x	Sì	Quando Lambda SnapStart è attivata `AWS_CONTAINER_CREDENTIALS_FULL_URI` e `AWS_CONTAINER_AUTHORIZATION_TOKEN` viene utilizzata automaticamente per l'autenticazione.
SDK per.NET 3.x	Sì	Quando Lambda SnapStart è attivata `AWS_CONTAINER_CREDENTIALS_FULL_URI` e `AWS_CONTA`

SDK	Su a	Note o ulteriori informazioni
		`INER_AUTHORIZATION_TOKEN` viene utilizzata automatic amente per l'autenticazione.
SDK per PHP 3.x	Sì	
SDK per Python (Boto3)	Sì	Quando Lambda SnapStart è attivata `AWS_CONTA INER_CREDENTIALS_FULL_URI` e `AWS_CONTA INER_AUTHORIZATION_TOKEN` viene utilizzata automatic amente per l'autenticazione.
SDK per Ruby 3.x	Sì	
SDK per Rust	Sì	
SDK per Swift	Sì	
Strumenti per V5 PowerShell	Sì	
Strumenti per PowerShell V4	Sì	

Provider di credenziali IAM Identity Center

> **Note**
>
> Per informazioni sulla comprensione del layout delle pagine delle impostazioni o sull'interpretazione della tabella Support by AWS SDKs and tools riportata di seguito, vedereInformazioni sulle pagine delle impostazioni di questa guida.

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

> **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 Come viene risolta l'autenticazione AWS SDKs e gli strumenti di IAM Identity Center

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

- (Consigliata) Configurazione del provider di token SSO: durate di sessione estese. Include il supporto per durate di sessione personalizzate.
- Configurazione legacy non aggiornabile: utilizza una sessione fissa di otto ore.

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

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

Per un'analisi approfondita su come gli strumenti SDKs e utilizzano e aggiornano le credenziali utilizzando questa configurazione, consulta. Come viene risolta l'autenticazione AWS SDKs e gli strumenti di IAM Identity Center

Prerequisiti

Devi prima abilitare IAM Identity Center. Per i dettagli sull'attivazione dell'autenticazione IAM Identity Center, consulta Enabling AWS IAM Identity Center nella AWS IAM Identity Center User Guide.

> **Note**
>
> In alternativa, per i prerequisiti completi e la necessaria configurazione dei `config` file condivisi, descritta in dettaglio in questa pagina, consulta le istruzioni guidate per la configurazioneUtilizzo di IAM Identity Center per autenticare AWS SDK e strumenti.

Configurazione del provider di token SSO

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 su questa sezione all'interno di un `config` file, vedere. Formato del file di configurazione

Il seguente esempio di `config` file condiviso configura l'SDK o lo strumento utilizzando un `dev` profilo per richiedere le credenziali IAM Identity Center.

```
[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
```

Gli esempi precedenti mostrano che è possibile definire una `sso-session` sezione e associarla a un profilo. 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. `sso_regionsso_start_url`, e `sso_registration_scopes` deve essere impostato all'interno della `sso-session` sezione.

`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. Questo tipo di autenticazione è uno schema di autenticazione HTTP che utilizza token di sicurezza noti come token di connessione. In questo scenario le impostazioni `sso_account_id` e `sso_role_name` non sono obbligatorie. Consulta la Servizio AWS guida individuale per determinare se il servizio supporta l'autorizzazione con 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. L'esempio precedente prevede `sso_registration_scopes` di fornire l'accesso necessario per elencare account e ruoli.

L'esempio seguente mostra come riutilizzare la stessa `sso-session` configurazione 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
```

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. Si consiglia 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 all'URL dell'emittente dello IAM Identity Center o all'URL del portale di accesso della tua organizzazione. Per ulteriori informazioni, 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, trova AWS l'URL del portale di accesso.

- In alternativa, a partire dalla versione 2.22.0 di AWS CLI, puoi invece utilizzare il valore per AWS Issuer URL.

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

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` Un'applicazione può richiedere uno o più ambiti e il token di accesso emesso all'applicazione è limitato agli ambiti consentiti. È `sso:account:access` necessario concedere un ambito minimo di per ottenere un token di aggiornamento dal servizio IAM Identity Center. Per l'elenco delle opzioni di ambito di accesso disponibili, consulta Access scopes nella Guida per l'AWS IAM Identity Center utente.

Questi ambiti definiscono le autorizzazioni richieste per l'autorizzazione per il client OIDC registrato e i token di accesso recuperati dal client. Gli ambiti autorizzano l'accesso agli endpoint autorizzati con token di connessione del Centro identità IAM.

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`

Support by AWS SDKs and tools

Di seguito sono SDKs supportate 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 per Java and the. AWS SDK per Kotlin

SDK	Su a	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	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 4.x	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.

SDK	Su a	Note o ulteriori informazioni
SDK per Swift	Sì	
Strumenti per V5 PowerShell	Sì	
Strumenti per PowerShell V4	Sì	

Provider di credenziali IMDS

> **Note**
>
> Per informazioni sulla comprensione del layout delle pagine delle impostazioni o sull'interpretazione della tabella Support by AWS SDKs and tools riportata di seguito, vedereInformazioni sulle pagine delle impostazioni di questa guida.

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 Work with instance metadata 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. SDKs 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 (IMDSv2) di Instance Metadata Service, 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), viene utilizzato come fallback. IMDSv1

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. IMDSv2

> **Note**
>
> I nuovi SDKs non supportano IMDSv1 e, quindi, non supportano questa impostazione. Per i dettagli, consulta la tabellaSupport by AWS SDKs and tools.

Valore predefinito: `false`.

Valori validi:

- **`true`**— Non utilizzare IMDSv1 come riserva.
- **`false`**— Utilizzare IMDSv1 come riserva.

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

L'endpoint di IMDS. Questo valore sostituisce la posizione predefinita in cui gli AWS SDK e gli strumenti cercheranno i metadati delle istanze Amazon EC2.

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. Comprendi la 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.

Support by AWS SDKs and tools

Di seguito sono SDKs supportate 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 per Java and the. AWS SDK per Kotlin

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

SDK	Su a	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	
SDK per Java 1.x	Parzia	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 4.x	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	Non utilizza il IMDSv1 fallback.
SDK per Swift	Sì	
Strumenti per V5 PowerShell	Sì	È possibile disabilitare il IMDSv1 fallback in modo esplicito nel codice utilizzando. `[Amazon.Util.EC2InstanceMetadata]::EC2MetadataV1Disabled = $true`

SDK	Su a	Note o ulteriori informazioni
Strumenti per V4 PowerShell	Sì	È possibile disabilitare il IMDSv1 fallback in modo esplicito nel codice utilizzando. `[Amazon.Util.EC2InstanceMetadata]::EC2MetadataV1Disabled = $true`

Provider di credenziali di processo

> **Note**
>
> Per informazioni sulla comprensione del layout delle pagine delle impostazioni o sull'interpretazione della tabella Support by AWS SDKs and tools riportata di seguito, vedereInformazioni sulle pagine delle impostazioni di questa guida.

SDKs forniscono un modo per estendere la catena di fornitori di credenziali per casi d'uso personalizzati. Questo provider può essere utilizzato per fornire implementazioni personalizzate, come il recupero delle credenziali da un archivio di credenziali locale o l'integrazione con il provider di identità locale.

Ad esempio, IAM Roles Anywhere utilizza per ottenere credenziali temporanee `credential_process` per conto dell'applicazione. Per configurarlo `credential_process` per questo utilizzo, consultaUtilizzo di IAM Roles Anywhere per l'autenticazione AWS SDKs e gli strumenti.

> **Note**
>
> Di seguito viene descritto un metodo di acquisizione delle credenziali da un processo esterno che potrebbe essere utilizzato se si esegue software all'esterno di. AWS Se stai basandoti su una risorsa di AWS calcolo, usa altri provider di credenziali. Se utilizzi questa opzione, devi assicurarti che il file di configurazione sia il più bloccato possibile utilizzando le migliori pratiche di sicurezza per il tuo sistema operativo. Verificate che lo strumento per le credenziali personalizzato non scriva alcuna informazione segretaStdErr, poiché è in AWS CLI grado di acquisire e registrare tali informazioni, esponendole potenzialmente a utenti non autorizzati. SDKs

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 a program/command 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. Comprendi la 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. Il percorso varia a seconda del sistema operativo:

  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"
  ```

- Può essere specificato all'interno di un profilo dedicato:

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

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 RFC3339 formattato. 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.

Support by AWS SDKs and tools

Di seguito sono SDKs supportate le funzionalità e le impostazioni descritte in questo argomento. Vengono annotate eventuali eccezioni parziali. Qualsiasi impostazione delle proprietà del sistema JVM è supportata solo da AWS SDK per Java and the. AWS SDK per Kotlin

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

SDK	Su a	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	
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 4.x	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 per Swift	Sì	
Strumenti per V5 PowerShell	Sì	
Strumenti per PowerShell V4	Sì	

AWS SDKs e funzionalità standardizzate di Tools

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

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

Argomenti

- Endpoint basati sull'account
- ID applicazione
- Metadati delle istanze Amazon EC2.
- Punti di accesso Amazon S3
- Punti di accesso multi-Regione di Amazon S3
- Autenticazione della sessione S3 Express One Zone
- Schema di autenticazione
- Regione AWS
- AWS STS Endpoint regionali
- Protezioni dell'integrità dei dati per Amazon S3
- Endpoint dual-stack e FIPS
- Rilevamento di endpoint
- Impostazioni generali di configurazione
- Iniezione del prefisso host
- Cliente IMDS
- Comportamento di ripetizione
- Richiesta di compressione
- Endpoint specifici del servizio
- Impostazioni predefinite di configurazione intelligente

Endpoint basati sull’account

> **Note**
>
> Per informazioni sulla comprensione del layout delle pagine delle impostazioni o sull'interpretazione della tabella Support by AWS SDKs and tools riportata di seguito, vedereInformazioni sulle pagine delle impostazioni di questa guida.

Gli endpoint basati su account aiutano a garantire prestazioni e scalabilità elevate utilizzando l' Account AWS ID per indirizzare le richieste di servizi che supportano questa funzionalità. Quando utilizzi un AWS SDK e un servizio che supportano gli endpoint basati su account, il client SDK crea e utilizza un endpoint basato sull'account anziché un endpoint regionale. Se l'ID dell'account non è visibile al client SDK, il client utilizzerà l'endpoint regionale. Gli endpoint basati sull'account assumono la forma di`https://<account-id>.ddb.<region>.amazonaws.com`, dove *`<account-id>`* e sono il tuo ID e*`<region>`*. Account AWS Regione AWS

Configura questa funzionalità utilizzando quanto segue:

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

L' Account AWS ID. Utilizzato per il routing degli endpoint basato sull'account. Un Account AWS ID ha un formato come 111122223333.

Il routing degli endpoint basato sull'account offre migliori prestazioni delle richieste per alcuni servizi.

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

Questa impostazione viene utilizzata per disattivare il routing degli endpoint basato sull'account, se necessario, e aggirare le regole basate sull'account.

Valore predefinito: `preferred`

Valori validi:

- **`preferred`**— L'endpoint deve includere l'ID dell'account, se disponibile.
- **`disabled`**: un endpoint risolto non include l’ID dell’account.

- **`required`**: l'endpoint deve includere l'ID dell'account. Se l'ID dell'account non è disponibile, l'SDK genera un errore.

Support by AWS SDKs and tools

Di seguito sono SDKs supportate 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 per Java and the. AWS SDK per Kotlin

SDK	Supp a	Rilasciato in versione SDK	Note o ulteriori informazioni
AWS CLI v2	Sì	2.25.0	
AWS CLI v1	Sì	1.38.0	
SDK per C++	No		
SDK per Go V2 (1.x)	Sì	v1.35.0	
SDK per Go 1.x (V1)	No		
SDK per Java 2.x	Sì	v2.28.4	
SDK per Java 1.x	Sì	v1.12.771	
SDK per 3.x JavaScript	Sì	v3.656.0	
SDK per 2.x JavaScript	No		
SDK per Kotlin	Sì	v1.3.37	
SDK per.NET 4.x	Sì	4.0.0	
SDK per.NET 3.x	No		

SDK	Supp a	Rilasciato in versione SDK	Note o ulteriori informazioni
SDK per PHP 3.x	Sì	v3.318.0	
SDK per Python (Boto3)	Sì	1.37.0	
SDK per Ruby 3.x	Sì	v1.123.0	
SDK per Rust	Sì	rilascio-2025-04-24	
SDK per Swift	Sì	1.2.0	
Strumenti per V5 PowerShell	No		
Strumenti per PowerShell V4	No		

ID applicazione

> **Note**
>
> Per informazioni sulla comprensione del layout delle pagine delle impostazioni o sull'interpretazione della tabella Support by AWS SDKs and tools riportata di seguito, vedereInformazioni sulle pagine delle impostazioni di questa guida.

Una singola Account AWS può essere utilizzata da più applicazioni clienti 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 SDKs e i servizi non utilizzano o interpretano questo valore se non per riemergere nelle comunicazioni con i clienti. Ad esempio,

questo valore può essere incluso nelle e-mail operative o nel codice Dashboard AWS Health 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, **`sdk.ua.appId`**- Proprietà del sistema JVM: solo Java/Kotlin

Questa impostazione è una stringa univoca che si assegna all'applicazione per identificare a quale delle applicazioni all'interno di un particolare dispositivo Account AWS effettua chiamate. 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.

Support by AWS SDKs and tools

Di seguito sono SDKs supportate 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 per Java and the. AWS SDK per Kotlin

SDK	Su a	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)	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	La proprietà del sistema JVM è. `aws.userAgentAppId`
SDK per.NET 4.x	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 per Swift	Sì	
Strumenti per V5 PowerShell	Sì	
Strumenti per PowerShell V4	Sì	

Metadati delle istanze Amazon EC2.

> **Note**
>
> Per informazioni sulla comprensione del layout delle pagine delle impostazioni o sull'interpretazione della tabella Support by AWS SDKs and tools riportata di seguito, vedereInformazioni sulle pagine delle impostazioni di questa guida.

Amazon EC2 fornisce un servizio su istanze chiamato Instance Metadata Service (IMDS). Per ulteriori informazioni su questo servizio, consulta Work with instance metadata 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
```

Support by AWS SDKs and tools

Di seguito sono SDKs supportate 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 per Java and the. AWS SDK per Kotlin

SDK	Su a	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	Parzia	Solo `AWS_METADATA_SERVICE_TIMEOUT` è supportata.
SDK per Java 1.x	Parzia	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 4.x	No	
SDK per.NET 3.x	No	

SDK	Su a	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	
SDK per Swift	No	
Strumenti per V5 PowerShell	No	
Strumenti per PowerShell V4	No	

Punti di accesso Amazon S3

> **Note**
>
> Per informazioni sulla comprensione del layout delle pagine delle impostazioni o sull'interpretazione della tabella Support by AWS SDKs and tools riportata di seguito, vedereInformazioni sulle pagine delle impostazioni di questa guida.

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 SDKs, puoi utilizzare il punto di accesso Amazon Resource Names (ARNs) 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 Amazon S3 e ARNs, 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 le configurazioni del client Regione AWS durante la costruzione dell'endpoint.

Support by AWS SDKs and tools

Di seguito sono SDKs supportate 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 per Java and the. AWS SDK per Kotlin

SDK	Su a	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	Su a	Note o ulteriori informazioni
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 4.x	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 per PHP 3.x	Sì	
SDK per Python (Boto3)	Sì	
SDK per Ruby 3.x	Sì	
SDK per Rust	No	
SDK per Swift	No	
Strumenti per V5 PowerShell	Sì	Non segue la precedenza standard; il valore `config` del file condiviso ha la precedenza sulla variabile di ambiente.
Strumenti per V4 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

> **Note**
>
> Per informazioni sulla comprensione del layout delle pagine delle impostazioni o sull'interpretazione della tabella Support by AWS SDKs and tools riportata di seguito, vedereInformazioni sulle pagine delle impostazioni di questa guida.

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 su Amazon Resource Names (ARNs) 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. AWS Librerie 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.

Support by AWS SDKs and tools

Di seguito sono SDKs supportate 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 per Java and the. AWS SDK per Kotlin

SDK	Supporta	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 per Kotlin	Sì	
SDK per.NET 4.x	Sì	
SDK per.NET 3.x	Sì	
SDK per PHP 3.x	Sì	
SDK per Python (Boto3)	Sì	

SDK	Su a	Note o ulteriori informazioni
SDK per Ruby 3.x	Sì	
SDK per Rust	Sì	
SDK per Swift	No	
Strumenti per V5 PowerShell	Sì	
Strumenti per PowerShell V4	Sì	

Autenticazione della sessione S3 Express One Zone

> **Note**
>
> Per informazioni sulla comprensione del layout delle pagine delle impostazioni o sull'interpretazione della tabella Support by AWS SDKs and tools riportata di seguito, vedereInformazioni sulle pagine delle impostazioni di questa guida.

S3 Express One Zone è la classe di storage ad alte prestazioni di Amazon S3 che offre una latenza di un millisecondo per i dati a cui si accede di frequente. Quando utilizzi S3 Express One Zone, i bucket AWS SDKs e gli strumenti utilizzano automaticamente l'autenticazione basata sulla sessione, ottimizzata per l'autorizzazione a bassa latenza delle richieste di dati. Utilizzi token di sessione con operazioni Zonal (a livello di oggetto) per distribuire la latenza associata all'autorizzazione su un numero di richieste in una sessione, riducendo il sovraccarico di autenticazione e migliorando le prestazioni complessive delle richieste.

I bucket S3 Express One Zone utilizzano un formato di denominazione specifico che include l'ID della zona di disponibilità, ad esempio. `bucket-name--usw2-az1--x-s3` Quando l'SDK rileva questo modello di denominazione, indirizza automaticamente le richieste agli endpoint S3 Express One Zone appropriati e applica il flusso di autenticazione ottimizzato. L'autenticazione della sessione crea credenziali temporanee specifiche per il bucket che forniscono un accesso a bassa latenza al bucket e vengono memorizzate nella cache e aggiornate automaticamente dall'SDK. Per ulteriori informazioni, consulta S3 Express One Zone nella Guida per l'utente di Amazon S3.

Per impostazione predefinita, l'autenticazione della sessione è abilitata per i bucket S3 Express One Zone.

Configura questa funzionalità utilizzando quanto segue:

`s3_disable_express_session_auth`- impostazione dei AWS **`config`** file condivisi, **`AWS_S3_DISABLE_EXPRESS_SESSION_AUTH`**- variabile d'ambiente, **`aws.disableS3ExpressAuth`**- Proprietà del sistema JVM: solo Java/Kotlin

Controlla se l'autenticazione della sessione S3 Express One Zone è disabilitata. Se impostato su`true`, l'SDK utilizza l'autenticazione SigV4 standard per i bucket S3 Express One Zone anziché l'autenticazione della sessione.

Valore predefinito: `false`

Valori validi:

- **`true`**— Disattiva l'autenticazione della sessione S3 Express One Zone.
- **`false`**— Abilita l'autenticazione della sessione S3 Express One Zone.

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

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

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

```
export AWS_S3_DISABLE_EXPRESS_SESSION_AUTH=true
```

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

```
setx AWS_S3_DISABLE_EXPRESS_SESSION_AUTH true
```

Support by AWS SDKs and tools

Di seguito sono SDKs supportate 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 per Java and the. AWS SDK per Kotlin

SDK	Supp a	Note o ulteriori informazioni
AWS CLI v2	Sì	
AWS CLI v1	Sì	
SDK per C++	Sì	
SDK per Go V2 (1.x)	Sì	
SDK per Go 1.x (V1)	No	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	No	
SDK per Kotlin	Sì	La proprietà del sistema JVM è. `aws.s3DisableExpressSessionAuth`
SDK per.NET 4.x	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 per Swift	Sì	
Strumenti per V5 PowerShell	Sì	

SDK	Supp a	Note o ulteriori informazioni
Strumenti per PowerShell V4	Sì	

Schema di autenticazione

> **Note**
>
> Per informazioni sulla comprensione del layout delle pagine delle impostazioni o sull'interpretazione della tabella Support by AWS SDKs and tools riportata di seguito, vedereInformazioni sulle pagine delle impostazioni di questa guida.

AWS i servizi supportano più schemi di autenticazione, come AWS Signature Version 4 (SigV4) e AWS Signature Version 4a (SigV4a). Per impostazione predefinita, SDKs seleziona gli schemi di autenticazione in base alle definizioni dei modelli di servizio e assegna priorità agli schemi che offrono la migliore compatibilità. Tuttavia, è possibile configurare lo schema di autenticazione preferito per ottimizzarlo in base a requisiti specifici.

A differenza di SigV4, le richieste firmate con SigV4a sono valide in più formati. Regioni AWS SigV4A offre una maggiore disponibilità attraverso la firma delle richieste tra regioni, che consente il failover automatico nelle regioni di backup durante le interruzioni regionali. Ciò è particolarmente vantaggioso per servizi globali come AWS Identity and Access Management Amazon CloudFront.

Per ulteriori informazioni su questi due schemi di autenticazione, consulta AWS Signature Version 4 per le richieste API nella IAM User Guide.

Configura questa funzionalità utilizzando quanto segue:

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

Specifica un elenco separato da virgole di schemi di autenticazione preferiti in ordine di priorità. Quando un servizio supporta più schemi di autenticazione, l'SDK tenta di utilizzare gli schemi di

questo elenco nell'ordine specificato, tornando al comportamento predefinito se nessuno degli schemi preferiti è disponibile.

Valore predefinito: Nessuno.

Valori validi: un elenco separato da virgole di uno o più dei seguenti elementi:

- **`sigv4`**— Signature Version 4 (prestazioni più elevate, regione singola)
- **`sigv4a`**— Signature Version 4a (disponibilità avanzata, supporto interregionale, offre prestazioni di firma più lente rispetto a SigV4)
- **`httpBearerAuth`**— Autenticazione con token HTTP Bearer

Gli spazi e i caratteri di tabulazione tra i nomi degli schemi vengono ignorati.

Esempio di impostazione di questo valore nel `config` file per preferire SigV4A:

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

`sigv4a_signing_region_set`- impostazione di file condivisi AWS **`config`**, **`AWS_SIGV4A_SIGNING_REGION_SET`**- variabile d'ambiente

Specifica un elenco separato da virgole Regioni AWS per la firma multiregionale SigV4A. Viene utilizzata come regione predefinita impostata per la richiesta se SigV4A è lo schema di autenticazione selezionato.

Valore predefinito: determinato dalla richiesta.

Valori validi: elenco separato da virgole di. Regioni AWS Gli spazi e i caratteri di tabulazione tra le regioni vengono ignorati.

Support by AWS SDKs and tools

Di seguito sono SDKs supportate 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 per Java and the. AWS SDK per Kotlin

SDK	Supporta	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 per Kotlin	Sì	
SDK per.NET 4.x	Sì	
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	Sì	
SDK per Swift	Sì	
Strumenti per V5 PowerShell	Sì	
Strumenti per PowerShell V4	No	

Regione AWS

> **Note**
>
> Per informazioni sulla comprensione del layout delle pagine delle impostazioni o sull'interpretazione della tabella Support by AWS SDKs and tools riportata di seguito, vedereInformazioni sulle pagine delle impostazioni di questa guida.

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 in modo programmatico Servizi AWS, SDKs usa un client class/object 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 nessuna regione è specificata esplicitamente per il client nel codice stesso, il client utilizza per impostazione predefinita la regione impostata tramite la seguente impostazione. `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 Regione AWS da utilizzare per le richieste. 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 SDKs ha 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.

Support by AWS SDKs and tools

Di seguito sono SDKs supportate 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 per Java and the. AWS SDK per Kotlin

SDK	Su a	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 per 2.x JavaScript	Sì	
SDK per Kotlin	Sì	
SDK per.NET 4.x	Sì	
SDK per.NET 3.x	Sì	

SDK	Si a	Note o ulteriori informazioni
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	
SDK per Swift	Sì	
Strumenti per V5 PowerShell	Sì	
Strumenti per PowerShell V4	Sì	

AWS STS Endpoint regionali

> **Note**
>
> Per informazioni sulla comprensione del layout delle pagine delle impostazioni o sull'interpretazione della tabella Support by AWS SDKs and tools che segue, vedereInformazioni sulle pagine delle impostazioni di questa guida.

AWS Security Token Service (AWS STS) è disponibile sia come servizio globale che regionale. Alcuni CLIs utilizzano AWS SDKs il servizio globale endpoint (`https://sts.amazonaws.com`) per impostazione predefinita, mentre altri utilizzano gli endpoint di servizio regionali ()`https://sts.{region_identifier}.{partition_domain}`. Nelle regioni abilitate per impostazione predefinita, le richieste all'endpoint AWS STS globale vengono servite automaticamente nella stessa regione da cui proviene la richiesta. Nelle regioni opt-in, le richieste all'endpoint AWS STS globale vengono servite da un'unica regione Regione AWS, gli Stati Uniti orientali (Virginia settentrionale). Per ulteriori informazioni sugli AWS STS endpoint, consulta Endpoints nell'AWS Security Token Service API Reference o Manage in an AWS STSRegione AWS nella User Guide.AWS Identity and Access Management

È consigliabile utilizzare gli endpoint regionali ogni volta che è possibile e configurare i propri. AWS Regione AWS I clienti con partizioni diverse da quelle commerciali devono utilizzare endpoint regionali. Non tutti SDKs gli strumenti supportano questa impostazione, ma tutti hanno un comportamento definito rispetto agli endpoint globali e regionali. Per ulteriori informazioni, consulta la sezione seguente.

> **Note**
>
> AWS ha apportato modifiche all'endpoint globale AWS Security Token Service (AWS STS`https://sts.amazonaws.com`) nelle Regioni abilitate di default per migliorarne la resilienza e le prestazioni. AWS STS le richieste all'endpoint globale vengono servite automaticamente Regione AWS come i tuoi carichi di lavoro. Queste modifiche non verranno implementate nelle Regioni con attivazione facoltativa. Ti consigliamo di utilizzare gli endpoint AWS STS regionali appropriati. Per ulteriori informazioni, consulta le modifiche AWS STS globali agli endpoint nella Guida per l'AWS Identity and Access Management utente.

Per SDKs gli strumenti che supportano questa impostazione, i clienti possono configurare la funzionalità utilizzando quanto segue:

`sts_regional_endpoints`- impostazione di 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:`regional`, vedi le eccezioni nella tabella seguente.

> **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 globale,`sts.amazonaws.com`.

- **`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'us`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
```

Support by AWS SDKs and tools

> **Note**
>
> È AWS consigliabile utilizzare gli endpoint regionali ogni volta che è possibile e configurare i propriRegione AWS.

La tabella che segue riassume, per il tuo SDK o strumento:

- Supporta l'impostazione: se sono supportate la variabile di `config` file condivisa e la variabile di ambiente per gli endpoint regionali STS.
- Valore di impostazione predefinito: il valore predefinito dell'impostazione, se supportata.
- Il client di servizio predefinito si rivolge a STS Endpoint: quale endpoint predefinito viene utilizzato dal client anche se l'impostazione per modificarlo non è disponibile.
- Comportamento di fallback del client di servizio: cosa fa l'SDK quando dovrebbe utilizzare un endpoint regionale ma non è stata configurata alcuna regione. Questo è il comportamento

indipendentemente dal fatto che stia utilizzando un endpoint regionale a causa di un valore predefinito o perché `regional` è stato selezionato dall'impostazione.

La tabella utilizza anche i seguenti valori:

- Endpoint globale:`https://sts.amazonaws.com`.
- Endpoint regionale: in base alla configurazione Regione AWS utilizzata dall'applicazione.
- **`us-east-1`**(Regionale): utilizza l'endpoint `us-east-1` Region ma con token di sessione più lunghi rispetto alle richieste globali tipiche.

SDK	Valore di impostazione predefinito	Il client di servizio predefinito è destinato a STS Endpoint	Comportamento di fallback del client di servizio	Note o ulteriori informazioni
AWS CLI v2	N/D	Endpoint regionale	Endpoint globale	
AWS CLI v1	`legacy`	Endpoint globale	Endpoint globale	
SDK per C++	N/D	Endpoint regionale	`us-east-1` (Regionale)	
SDK per Go V2 (1.x)	N/D	Endpoint regionale	Richiesta non riuscita	
SDK per Go 1.x (V1)	`legacy`	Endpoint globale	Endpoint globale	Per utilizzare le impostazioni dei `config` file condivisi, devi attivare il caricamento dal file di configurazione; vedi Sessioni.
SDK per Java 2.x	N/D	Endpoint regionale	Richiesta non riuscita	Se non è configurata alcuna regione, `AssumeRole` e

SDK	:	Valore di impostazione predefinito	Il client di servizio predefinito è destinato a STS Endpoint	Comportamento di fallback del client di servizio	Note o ulteriori informazioni
					`AssumeRoleWithWebIdentity` utilizzerà l'endpoint STS globale.
SDK per Java 1.x	S	`legacy`	Endpoint globale	Endpoint globale	
SDK per 3.x JavaScript	N	N/D	Endpoint regionale	`us-east-1` (Regionale)	
SDK per 2.x JavaScript	S	`legacy`	Endpoint globale	Endpoint globale	
SDK per Kotlin	N	N/D	Endpoint regionale	Endpoint globale	
SDK per.NET 4.x	N	N/D	Endpoint regionale	`us-east-1` (Regionale)	
SDK per.NET 3.x	S	`regional`	Endpoint globale	Endpoint globale	
SDK per PHP 3.x	S	`regional`	Endpoint globale	Richiesta non riuscita	
SDK per Python (Boto3)	S	`regional`	Endpoint globale	Endpoint globale	
SDK per Ruby 3.x	S	`regional`	Endpoint regionale	Richiesta fallita	

SDK	:	Valore di impostazione predefinito	Il client di servizio predefinito è destinato a STS Endpoint	Comportamento di fallback del client di servizio	Note o ulteriori informazioni
SDK per Rust	N	N/D	Endpoint regionale	Richiesta fallita	
SDK per Swift	N	N/D	Endpoint regionale	Richiesta non riuscita	
Strumenti per PowerShell V5	S	`regional`	Endpoint globale	Endpoint globale	
Strumenti per PowerShell V4	S	`regional`	Endpoint globale	Endpoint globale	

Protezioni dell'integrità dei dati per Amazon S3

> **Note**
>
> Per informazioni sulla comprensione del layout delle pagine delle impostazioni o sull'interpretazione della tabella Support by AWS SDKs and tools riportata di seguito, vedereInformazioni sulle pagine delle impostazioni di questa guida.

Da qualche tempo, AWS SDKs supportiamo i controlli di integrità dei dati durante il caricamento o il download di dati da Amazon Simple Storage Service. In precedenza, questi controlli erano facoltativi. Ora, abbiamo abilitato questi controlli per impostazione predefinita, utilizzando algoritmi basati su CRC come o NVME. CRC32 CRC64 Sebbene ogni SDK o strumento abbia un algoritmo predefinito, puoi scegliere un algoritmo diverso. Se lo desideri, puoi anche continuare a fornire manualmente un checksum precalcolato per i caricamenti. Il comportamento coerente tra i caricamenti, i caricamenti multiparte, i download e le modalità di crittografia semplifica i controlli di integrità lato client.

Le versioni più recenti del nostro programma AWS SDKs calcola AWS CLI automaticamente un checksum basato sul controllo di ridondanza ciclica (CRC) per ogni caricamento e lo invia ad Amazon S3. Amazon S3 calcola in modo indipendente un checksum sul lato server e lo convalida rispetto al valore fornito prima di archiviare in modo duraturo l'oggetto e il relativo checksum nei metadati dell'oggetto. Memorizzando il checksum nei metadati insieme all'oggetto, quando l'oggetto viene scaricato, lo stesso checksum può essere restituito automaticamente e utilizzato anche per convalidare i download. È inoltre possibile verificare il checksum memorizzato nei metadati dell'oggetto in qualsiasi momento.

Per ulteriori informazioni sulle operazioni di checksum, sui caricamenti in più parti o sull'elenco degli algoritmi di checksum supportati, consulta Checking object integrity in Amazon S3 nella Amazon Simple Storage Service User Guide.

Caricamenti in più parti:

Amazon S3 fornisce inoltre agli sviluppatori checksum completi di oggetti coerenti per caricamenti singoli e multiparte.

Quando carichi file in più parti, calcola i checksum per ogni parte SDKs . Amazon S3 utilizza questi checksum per verificare l'integrità di ogni parte tramite l'API. `UploadPart` Inoltre, Amazon S3 convalida la dimensione e il checksum dell'intero file quando chiami l'API. `CompleteMultipartUpload`

Se il tuo SDK dispone di un Amazon S3 Transfer Manager per facilitare i caricamenti in più parti, i checksum vengono convalidati per le parti utilizzando l'algoritmo predefinito specifico dell'SDK riportato nella tabella. Support by AWS SDKs and tools Puoi attivare un checksum completo dell'oggetto impostando l'impostazione o scegliendo di utilizzare l'algoritmo NVME. `checksum_type` `FULL_OBJECT` CRC64

Se utilizzi una versione precedente di SDK o: AWS CLI

Se l'applicazione utilizza una versione precedente a dicembre 2024 dell'SDK o dello strumento, Amazon S3 calcola comunque CRC64 un checksum NVME sui nuovi oggetti e lo archivia nei metadati degli oggetti per riferimenti futuri. Successivamente puoi confrontare il CRC memorizzato con un CRC calcolato da parte tua e verificare che la trasmissione di rete sia stata corretta. Inoltre, potete comunque estendere manualmente la protezione dell'integrità fornendo checksum precalcolati con le vostre `UploadPart`richieste `PutObject`or, che è la tecnica standard per risolvere questo problema nelle versioni precedenti.

Configura questa funzionalità utilizzando quanto segue:

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

Per impostazione predefinita, gli utenti scelgono di calcolare il checksum di una richiesta quando inviano una richiesta. L'utente può scegliere uno qualsiasi degli algoritmi di checksum disponibili come parte della creazione della richiesta. In caso contrario, viene utilizzato un algoritmo predefinito specifico dell'SDK. Consulta la Support by AWS SDKs and tools tabella per l'algoritmo predefinito per ogni SDK o strumento.

Valore predefinito: `WHEN_SUPPORTED`

Valori validi:

- **`WHEN_SUPPORTED`**— La convalida del checksum viene eseguita su tutti i payload della richiesta se supportata dal funzionamento dell'API, come i trasferimenti di dati su Amazon S3.
- **`WHEN_REQUIRED`**— La convalida del checksum viene eseguita solo quando richiesta dal funzionamento dell'API.

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

Per impostazione predefinita, gli utenti accettano la convalida del checksum della risposta quando inviano una richiesta. Viene calcolato un checksum per il payload della risposta e confrontato con l'intestazione della risposta checksum. Se la convalida del checksum fallisce, viene segnalato un errore all'utente durante la lettura del payload.

L'intestazione della risposta checksum indica anche l'algoritmo per il checksum. Il client Amazon S3 tenta di convalidare i checksum di risposta per tutte le operazioni API Amazon S3 che supportano i checksum. Tuttavia, se l'SDK non ha implementato l'algoritmo di checksum specificato, questa convalida viene ignorata.

Valore predefinito: `WHEN_SUPPORTED`

Valori validi:

- **`WHEN_SUPPORTED`**— La convalida del checksum viene eseguita su tutti i payload di risposta se supportata dal funzionamento dell'API, come i trasferimenti di dati verso Amazon S3.
- **`WHEN_REQUIRED`**— La convalida del checksum viene eseguita solo se supportata dall'operazione API e il chiamante ha abilitato esplicitamente il checksum per l'operazione. Ad

esempio, quando viene chiamata l'`GetObject`API Amazon S3 e il `ChecksumMode` parametro è impostato su enabled.

Support by AWS SDKs and tools

Di seguito sono SDKs supportate le funzionalità e le impostazioni descritte in questo argomento. Vengono annotate eventuali eccezioni parziali. Qualsiasi impostazione delle proprietà del sistema JVM è supportata solo da AWS SDK per Java and the. AWS SDK per Kotlin

> **Note**
>
> Nella tabella seguente, «CRT» si riferisce al progetto AWS Librerie Common Runtime (CRT) e potrebbe richiedere l'aggiunta di una dipendenza aggiuntiva.

SDK	Suppo a	Algoritmo di checksum predefinito	Algoritmi di checksum supportati	Note o ulteriori informazioni
AWS CLI v2	Sì	CRC64NVME	CRC64NVME, C, CRC32 CRC32 SHA1 SHA256	Per la AWS CLI v1, l'algoritmo predefinito e gli algoritmi supportati saranno identici a Python (Boto3).
SDK per C++	Sì	CRC64NVME	CRC64NVME, C, CRC32 CRC32 SHA1 SHA256	
SDK per Go V2 (1.x)	Sì	CRC32	CRC64NVME, C, CRC32 CRC32 SHA1 SHA256	
SDK per Go 1.x (V1)	No			
SDK per Java 2.x	Sì	CRC32	CRC64NVME (solo tramite CRT), C,,	

SDK	Suppo a	Algoritmo di checksum predefinito	Algoritmi di checksum supportati	Note o ulteriori informazioni
			CRC32 CRC32 SHA1 SHA256	
SDK per Java 1.x	No			
SDK per 3.x JavaScript	Sì	CRC32	CRC32, CRC32 C, SHA1 SHA256	
SDK per 2.x JavaScript	No			
SDK per Kotlin	Sì	CRC32	CRC32, C, CRC32 SHA1 SHA256	
SDK per.NET 4.x	Sì	CRC32	CRC32, CRC32 C, SHA1 SHA256	
SDK per.NET 3.x	Sì	CRC32	CRC32, CRC32 C, SHA1 SHA256	
SDK per PHP 3.x	Sì	CRC32	CRC32, CRC32 C (solo tramite CRT), SHA1 SHA256	`awscrt`l'estensione è necessaria per utilizzare C. CRC32
SDK per Python (Boto3)	Sì	CRC32	CRC64NVME (solo tramite CRT), CRC32 C (solo tramite CRT) CRC32, SHA1 SHA256	

SDK	Suppo a	Algoritmo di checksum predefinito	Algoritmi di checksum supportati	Note o ulteriori informazioni
SDK per Ruby 3.x	Sì	CRC32	CRC64NVME (solo tramite CRT), CRC32 C (solo tramite CRT) CRC32, SHA1 SHA256	
SDK per Rust	Sì	CRC32	CRC64NVME, C CRC32, CRC32 SHA1 SHA256	
SDK per Swift	Sì	CRC32	CRC64NVME, C, CRC32 CRC32 SHA1 SHA256	Dipendenza CRT richiesta per tutti gli algoritmi.
Strumenti per V5 PowerShell	Sì	CRC32	CRC32, CRC32 C, SHA1 SHA256	
Strumenti per PowerShell V4	Sì	CRC32	CRC32, CRC32 C, SHA1 SHA256	

Endpoint dual-stack e FIPS

> **Note**
>
> Per informazioni sulla comprensione del layout delle pagine delle impostazioni o sull'interpretazione della tabella Support by AWS SDKs and tools riportata di seguito, vedereInformazioni sulle pagine delle impostazioni di questa guida.

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 che il IPv6 traffico, consulta Using IPv4 Using Amazon S3 dual-stack endpoint nella Amazon Simple Storage Service User Guide. Gli endpoint dual-stack sono disponibili per determinati 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 convalidata secondo FIPS 140. 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 and 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.

Support by AWS SDKs and tools

Di seguito sono SDKs supportate 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 per Java and the. AWS SDK per Kotlin

SDK	Supporta	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 4.x	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 a	Note o ulteriori informazioni
SDK per Rust	Sì	
SDK per Swift	Sì	
Strumenti per V5 PowerShell	Sì	
Strumenti per PowerShell V4	Sì	

Rilevamento di endpoint

> **Note**
>
> Per informazioni sulla comprensione del layout delle pagine delle impostazioni o sull'interpretazione della tabella Support by AWS SDKs and tools riportata di seguito, vedereInformazioni sulle pagine delle impostazioni di questa guida.

SDKs utilizza endpoint discovery per accedere agli endpoint del servizio (URLs per accedere a varie risorse), pur mantenendo la flessibilità necessaria per apportare modifiche URLs in base AWS alle esigenze. 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.

Support by AWS SDKs and tools

Di seguito sono SDKs supportate 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 per Java and the. AWS SDK per Kotlin

SDK	Su a	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	Su a	Note o ulteriori informazioni
SDK per.NET 4.x	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	Supportato solo per Timestream.
SDK per Swift	No	
Strumenti per V5 PowerShell	Sì	
Strumenti per PowerShell V4	Sì	

Impostazioni generali di configurazione

> **Note**
>
> Per informazioni sulla comprensione del layout delle pagine delle impostazioni o sull'interpretazione della tabella Support by AWS SDKs and tools che segue, vedereInformazioni sulle pagine delle impostazioni di questa guida.

SDKs supporta 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 SSL/TLS connessioni.

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
```

A causa delle differenze nel modo in cui i sistemi operativi gestiscono i percorsi e l'escape dei caratteri di percorso, il seguente è un esempio di impostazione di questo valore nel `config` file in Windows:

```
[default]
ca_bundle = C:\\Users\\username\\.aws\\aws-custom-bundle.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 in AWS CLI AWS SDKs e altri strumenti.

Valore predefinito: `json`

Valori validi:

- **`json`**: l'output è formattato come una stringa JSON.
- **`yaml`**: l'output è formattato come una stringa YAML.
- **`yaml-stream`**: l'output viene inviato in streaming e formattato come una 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. Questa formattazione può essere utile per passare l'output a un elaboratore di testi, ad esempio `grep`, `sed` o `awk`.
- **`table`**: l'output è formattato come una tabella in cui si utilizzano 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 dei 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.

Support by AWS SDKs and tools

Di seguito sono SDKs supportate le funzionalità e le impostazioni descritte in questo argomento. Vengono annotate eventuali eccezioni parziali. Qualsiasi impostazione delle proprietà del sistema JVM è supportata solo da AWS SDK per Java and the. AWS SDK per Kotlin

SDK	Supporta	Note o ulteriori informazioni
AWS CLI v2	Parziale	`api_versions` non supportato.
SDK per C++	Sì	
SDK per Go V2 (1.x)	Parziale	`api_versions` e `parameter_validation` non supportato.
SDK per Go 1.x (V1)	Parziale	`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 per Java 1.x	No	
SDK per 3.x JavaScript	Sì	
SDK per 2.x JavaScript	Sì	
SDK per Kotlin	No	
SDK per.NET 4.x	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	Su a	Note o ulteriori informazioni
SDK per Rust	No	
SDK per Swift	No	
Strumenti per V5 PowerShell	No	
Strumenti per PowerShell V4	No	

Iniezione del prefisso host

> **Note**
>
> Per informazioni sulla comprensione del layout delle pagine delle impostazioni o sull'interpretazione della tabella Support by AWS SDKs and tools riportata di seguito, vedereInformazioni sulle pagine delle impostazioni di questa guida.

L'iniezione del prefisso host è una funzionalità che consente di anteporre AWS SDKs automaticamente un prefisso al nome host degli endpoint del servizio per determinate operazioni API. Questo prefisso può essere una stringa statica o un valore dinamico che include i dati dei parametri della richiesta.

Ad esempio, quando si utilizza Amazon Simple Storage Service per eseguire azioni su oggetti o bucket Amazon S3, l'SDK sostituisce il nome e l' Account AWS ID del bucket nell'endpoint API finale.

Sebbene questo comportamento sia necessario per i normali endpoint di AWS servizio, può causare problemi quando si utilizzano endpoint personalizzati come endpoint VPC o strumenti di test locali. In questi casi, potrebbe essere necessario disabilitare l'iniezione del prefisso dell'host.

Configura questa funzionalità utilizzando quanto segue:

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

Questa impostazione controlla se l'SDK o lo strumento modificheranno il nome host dell'endpoint anteponendo un prefisso host come definito nell'oggetto o nella variabile client dell'SDK.

Valore predefinito: `false`

Valori validi:

- **true**— Disabilita l'iniezione del prefisso dell'host. L'SDK non modificherà il nome host dell'endpoint.
- **false**— Abilita l'iniezione del prefisso dell'host. L'SDK aggiungerà il prefisso dell'host al nome host dell'endpoint.

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

```
[default]
disable_host_prefix_injection = true
```

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

```
export AWS_DISABLE_HOST_PREFIX_INJECTION=true
```

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

```
setx AWS_DISABLE_HOST_PREFIX_INJECTION true
```

Esempi di iniezione del prefisso host

La seguente tabella di esempi mostra come SDKs modificare l'endpoint finale quando l'iniezione del prefisso host è abilitata e disabilitata.

- Prefisso host: il modello della stringa di proprietà del prefisso host impostata nell'oggetto client o nella variabile del codice dell'SDK.
- Ingressi: input aggiuntivi impostati nell'oggetto o nella variabile client dell'SDK nel codice.
- Endpoint client: l'endpoint derivato dal client.
- Valore dell'impostazione: valore risolto per l'impostazione precedente.

- Endpoint risultante: l'endpoint risultante utilizzato dal client SDK per effettuare la chiamata API.

Prefisso dell'host	Input	Endpoint del client	Valore di impostazione	Endpoint risultante
«dati».	{}	"https://service.us-west-2.amazonaws.com"	false	"https://data.service.us-west-2.amazonaws.com"
«{Bucket} - {AccountId}.»	Secchio: «amzn-s3-demo-bucket1",:" 123456789012" AccountId	"https://service.us-west-2.amazonaws.com"	false	"https://amzn-s3-demo-bucket1-123456789012.service.us-west-2.amazonaws.com"
«dati».	{}	"https://override.us-west-2.amazonaws.com"(come endpoint sostitutivo)	true	"https://override.us-west-2.amazonaws.com"

Support by AWS SDKs and tools

Di seguito sono SDKs supportate 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 per Java and the. AWS SDK per Kotlin

SDK	Supporta	Note o ulteriori informazioni
AWS CLI v2	Sì	

SDK	Su a	Note o ulteriori informazioni
SDK per C++	No	Impostazione non supportata, ma può essere configurata nel codice sul client utilizzando:. `enableHostPrefixInjection`
SDK per Go V2 (1.x)	No	Può essere disabilitato utilizzando il middleware.
SDK per Go 1.x (V1)	No	
SDK per Java 2.x	No	Impostazione non supportata, ma può essere configurata nel codice sul client utilizzando:. `SdkAdvancedClientOption.DISABLE_HOST_PREFIX_INJECTION`
SDK per Java 1.x	No	Impostazione non supportata, ma può essere configurata nel codice sul client utilizzando:. `withDisableHostPrefixInjection`
SDK per 3.x JavaScript	No	Impostazione non supportata, ma può essere configurata nel codice sul client utilizzando:. `disableHostPrefix`
SDK per 2.x JavaScript	No	L'impostazione non è supportata, ma può essere configurata nel codice sul client utilizzando:. `hostPrefixEnabled`
SDK per Kotlin	No	
SDK per.NET 4.x	No	Impostazione non supportata, ma può essere configurata nel codice sul client utilizzando:. `DisableHostPrefixInjection`
SDK per.NET 3.x	No	Impostazione non supportata, ma può essere configurata nel codice sul client utilizzando:. `DisableHostPrefixInjection`
SDK per PHP 3.x	No	Impostazione non supportata, ma può essere configurata nel codice sul client utilizzando:. `disable_host_prefix_injection`

SDK	Su a	Note o ulteriori informazioni
SDK per Python (Boto3)	Sì	Può essere configurato in codice sul client utilizzando:. `inject_host_prefix`
SDK per Ruby 3.x	No	Impostazione non supportata, ma può essere configurata nel codice sul client utilizzando:. `disable_host_prefix_injection`
SDK per Rust	No	
SDK per Swift	No	
Strumenti per V5 PowerShell	No	L'impostazione non è supportata, ma può essere inclusa in cmdlet specifici utilizzando il parametro. `-ClientConfig @{DisableHostPrefixInjection = $true}`
Strumenti per V4 PowerShell	No	L'impostazione non è supportata, ma può essere inclusa in cmdlet specifici utilizzando il parametro. `-ClientConfig @{DisableHostPrefixInjection = $true}`

Cliente IMDS

> **Note**
>
> Per informazioni sulla comprensione del layout delle pagine delle impostazioni o sull'interpretazione della tabella Support by AWS SDKs and tools riportata di seguito, vedereInformazioni sulle pagine delle impostazioni di questa guida.

SDKs implementa un client Instance Metadata Service versione 2 (IMDSv2) utilizzando richieste orientate alla sessione. Per ulteriori informazioni IMDSv2, consulta Use IMDSv2 in the 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 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 SDKs. 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 I/O flusso 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.

Support by AWS SDKs and tools

Di seguito sono SDKs supportate 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 per Java and the. AWS SDK per Kotlin

SDK	Supporta	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)	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	No	
SDK per.NET 4.x	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 per Swift	Sì	
Strumenti per V5 PowerShell	Sì	
Strumenti per PowerShell V4	Sì	

Comportamento di ripetizione

> **Note**
>
> Per informazioni sulla comprensione del layout delle pagine delle impostazioni o sull'interpretazione della tabella Support by AWS SDKs and tools riportata di seguito, vedereInformazioni sulle pagine delle impostazioni di questa guida.

Il comportamento Riprova include impostazioni relative al modo in cui il SDKs tentativo di ripristino degli errori risultanti dalle richieste effettuate a. Servizi AWS

Configura questa funzionalità utilizzando quanto segue:

`retry_mode`- impostazione dei AWS **`config`** file condivisi, **`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: questo valore è specifico del tuo SDK. Controlla la tua guida SDK specifica o la base di codice del tuo SDK per verificare se è predefinita. `retry_mode`

Valori validi:

- `standard`— (Consigliato) Il set consigliato di regole per i tentativi ripetuti. AWS SDKs Questa modalità include un set standard di errori che vengono ripetuti e regola automaticamente il numero di tentativi per massimizzare la disponibilità e la stabilità. Questa modalità è sicura per l'uso in applicazioni multi-tenant. 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 di riprova, adatta solo per casi d'uso specializzati, che include la funzionalità della modalità standard e la limitazione automatica della velocità lato client. Questa modalità di riprova non è consigliata per le applicazioni multi-tenant, a meno che non si prenda cura di isolare i tenant delle applicazioni. Per ulteriori informazioni, consulta Scelta tra le `standard` modalità e `adaptive` riprova. Questa modalità è sperimentale e potrebbe modificare il comportamento in futuro.
- `legacy`— (Non consigliato) Specifico per il tuo SDK (consulta la guida SDK specifica o la base di codice dell'SDK).

`max_attempts`- impostazione di file condivisi AWS **`config`**, **`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.

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.

Standard	Adattabile
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 codice deve essere progettato per avere un client per coppia. Region-and-table 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

AWS SDKs Utilizzano i token bucket per decidere se ritentare una richiesta e (nel caso della modalità `adaptive` retry) con quale rapidità inviare le richieste. L'SDK utilizza due bucket di token: un bucket di token retry e un bucket di token di frequenza di richiesta.

- Il bucket retry token viene utilizzato per determinare se l'SDK debba disabilitare temporaneamente i nuovi tentativi per proteggere i servizi upstream e downstream durante le interruzioni. I token vengono acquisiti dal bucket prima di tentare di riprovare e i token vengono restituiti al bucket quando le richieste hanno esito positivo. Se il bucket è vuoto quando viene tentato un nuovo tentativo, l'SDK non riproverà a eseguire la richiesta.

- Il bucket del token del tasso di richiesta viene utilizzato solo in modalità di `adaptive` ripetizione per determinare la velocità di invio delle richieste. I token vengono acquisiti dal bucket prima dell'invio di una richiesta e i token vengono restituiti al bucket a una velocità determinata dinamicamente in base alle risposte di limitazione restituite dal servizio.

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`:

Questo passaggio viene utilizzato solo in `adaptive` modalità riprova. Questo passaggio acquisisce un token dal bucket di token del tasso di richiesta. Se un token non è disponibile, aspetterà che ne diventi disponibile uno. Il tuo SDK potrebbe avere opzioni di configurazione disponibili per fallire la richiesta anziché attendere. I token presenti nel bucket vengono ricaricati a una velocità determinata dinamicamente, in base al numero di risposte di limitazione ricevute dal client.

`SendHTTPRequest`:

Questo passaggio invia la richiesta a. AWS La maggior parte AWS SDKs utilizza una libreria HTTP che utilizza pool di connessioni per riutilizzare una connessione esistente quando si effettua una

richiesta HTTP. In genere, le connessioni vengono riutilizzate se una richiesta non è riuscita a causa di errori di limitazione, ma non se una richiesta fallisce a causa di un errore temporaneo.

`RequestBookkeeping`:

I token vengono aggiunti al token bucket se la richiesta ha esito positivo. Solo per la modalità `adaptive` riprova, la frequenza di riempimento del bucket di token del tasso di richiesta 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`:

Il numero massimo di tentativi predefinito è impostato dall'`retry_mode`impostazione, a meno che l'impostazione non venga sovrascritta dall'impostazione. `max_attempts`

`HasRetryQuota`

Questo passaggio acquisisce un token dal bucket retry token. Se il bucket del token Retry è vuoto, la richiesta non verrà ritentata.

`ExponentialBackoff`

Per un errore che può essere ritentato, il ritardo tra i tentativi viene calcolato utilizzando un backoff esponenziale troncato. Viene SDKs utilizzato 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 SDKs. Consulta la guida SDK o il codice sorgente specifici per avere conferma.

Support by AWS SDKs and tools

Di seguito sono SDKs supportate 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 per Java and the. AWS SDK per Kotlin

SDK	Su a	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	Si a	Note o ulteriori informazioni
SDK per Kotlin	Sì	
SDK per.NET 4.x	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 per Swift	Sì	
Strumenti per V5 PowerShell	Sì	
Strumenti per PowerShell V4	Sì	

Richiesta di compressione

> **Note**
>
> Per informazioni sulla comprensione del layout delle pagine delle impostazioni o sull'interpretazione della tabella Support by AWS SDKs and tools riportata di seguito, vedereInformazioni sulle pagine delle impostazioni di questa guida.

AWS SDKs e gli strumenti possono comprimere automaticamente i payload quando inviano richieste a Servizi AWS quel supporto che riceve payload compressi. 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.

Support by AWS SDKs and tools

Di seguito sono SDKs supportate 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 per Java and the. AWS SDK per Kotlin

SDK	Supporta	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 per Kotlin	Sì	
SDK per.NET 4.x	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 per Swift	No	
Strumenti per V5 PowerShell	Sì	
Strumenti per PowerShell V4	Sì	

Endpoint specifici del servizio

> **Note**
>
> Per informazioni sulla comprensione del layout delle pagine delle impostazioni o sull'interpretazione della tabella Support by AWS SDKs and tools riportata di seguito, vedereInformazioni sulle pagine delle impostazioni di questa guida.

La configurazione degli endpoint specifica del 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 dei 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 l'elenco di tutte le variabili di ambiente specifiche del servizio, consulta 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 `AWS_ENDPOINT_URL` per questo servizio.

```
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 l'elenco di tutte le variabili di ambiente specifiche del servizio, consulta Identificatori per endpoint specifici del servizio.

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

Configura 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 sezione `services` nel file `config` 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 l'endpoint specifico del servizio per Amazon URLs S3 e AWS Elastic Beanstalk

nello stesso profilo. 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 sezione `services`, consulta 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 dispone di credenziali basate sui ruoli configurate tramite un parametro `source_profile` per consentire a IAM di assumere la funzionalità del ruolo, l'SDK utilizza solo le configurazioni di servizio relative al profilo specificato. Non utilizza profili a cui sono associati ruoli concatenati. Ad esempio, può utilizzare il seguente file `config` 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 utilizzi il profilo `B` ed effettui una chiamata nel codice verso Amazon EC2, l'endpoint viene risolto 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 viene risolto come l'endpoint globale definito nel profilo `A`. Affinché un endpoint globale abbia effetto sul profilo `B`, è necessario impostare `endpoint_url` 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.

Support by AWS SDKs and tools

Di seguito sono SDKs supportate 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 per Java and the. AWS SDK per Kotlin

SDK	Supporta	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 per Kotlin	Sì	
SDK per.NET 4.x	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 per Swift	Sì	

SDK	Su a	Note o ulteriori informazioni
Strumenti per V5 PowerShell	Sì	
Strumenti per PowerShell V4	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`

serviceId	Cl id at de se pe il fil cc A\ **cc**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
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
ApiGatewayManagementApi	ap yn nt	AWS_ENDPOINT_URL_APIGATEWAYMANAGEMENTAPI
ApiGatewayV2	ap y\	AWS_ENDPOINT_URL_APIGATEWAYV2
AppConfig	ap	AWS_ENDPOINT_URL_APPCONFIG
AppConfigData	ap da	AWS_ENDPOINT_URL_APPCONFIGDATA
AppFabric	ap	AWS_ENDPOINT_URL_APPFABRIC

serviceId	Cl id at de se pe il fil co A\ **co**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
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
Application Insights	ap or ts	AWS_ENDPOINT_URL_APPLICATION_INSIGHTS
ApplicationCostPro filer	ap or f:	AWS_ENDPOINT_URL_APPLICATIONCOSTPROFILER
App Mesh	ap	AWS_ENDPOINT_URL_APP_MESH
AppRunner	ap	AWS_ENDPOINT_URL_APPRUNNER
AppStream	ap	AWS_ENDPOINT_URL_APPSTREAM
AppSync	ap	AWS_ENDPOINT_URL_APPSYNC

serviceId	Cl id at de se pe il fil cc A\ **ce**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
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
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

serviceId	Cl id at de se pe il fil cc A\ **cc**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
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
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	ce ce	AWS_ENDPOINT_URL_COST_EXPLORER

serviceId	Cl id at de se pe il fil cc A\ **ce**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
chatbot	ch	AWS_ENDPOINT_URL_CHATBOT
Chime	ch	AWS_ENDPOINT_URL_CHIME
Chime SDK Identity	ch _:	AWS_ENDPOINT_URL_CHIME_SDK_IDENTITY
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	cl s	AWS_ENDPOINT_URL_CLEANROOMS

serviceId	Cl id at de se pe il fil cc A\ **cc**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
CleanRoomsML	c: sr	AWS_ENDPOINT_URL_CLEANROOMSML
Cloud9	c:	AWS_ENDPOINT_URL_CLOUD9
CloudControl	c: rc	AWS_ENDPOINT_URL_CLOUDCONTROL
CloudDirectory	c: ct	AWS_ENDPOINT_URL_CLOUDDIRECTORY
CloudFormation	c: af	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

serviceId	Cl id at de se pe il fil cc A\ **cc**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
CloudSearch	c: ch	AWS_ENDPOINT_URL_CLOUDSEARCH
CloudSearch Domain	c: ch	AWS_ENDPOINT_URL_CLOUDSEARCH_DOMAIN
CloudTrail	c: l	AWS_ENDPOINT_URL_CLOUDTRAIL
CloudTrail Data	c: l_	AWS_ENDPOINT_URL_CLOUDTRAIL_DATA
CloudWatch	c: h	AWS_ENDPOINT_URL_CLOUDWATCH
codeartifact	cc ac	AWS_ENDPOINT_URL_CODEARTIFACT
CodeBuild	cc	AWS_ENDPOINT_URL_CODEBUILD
CodeCatalyst	cc ys	AWS_ENDPOINT_URL_CODECATALYST

serviceId	Cl id at de se pe il fil co AV **co**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
CodeCommit	co t	AWS_ENDPOINT_URL_CODECOMMIT
CodeDeploy	co y	AWS_ENDPOINT_URL_CODEDEPLOY
CodeGuru Reviewer	co re	AWS_ENDPOINT_URL_CODEGURU_REVIEWER
CodeGuru Security	co se	AWS_ENDPOINT_URL_CODEGURU_SECURITY
CodeGuruProfiler	co ro	AWS_ENDPOINT_URL_CODEGURUPROFILER
CodePipeline	co in	AWS_ENDPOINT_URL_CODEPIPELINE
CodeStar	co	AWS_ENDPOINT_URL_CODESTAR
CodeStar connections	co co ns	AWS_ENDPOINT_URL_CODESTAR_CONNECTIONS

serviceId	Cl id at de se pe il fil co A\ **co**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
codestar notifications	co no io	AWS_ENDPOINT_URL_CODESTAR_NOTIFICATIONS
Cognito Identity	co de	AWS_ENDPOINT_URL_COGNITO_IDENTITY
Cognito Identity Provider	co de ro	AWS_ENDPOINT_URL_COGNITO_IDENTITY_PROVIDER
Cognito Sync	co 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

serviceId	Cl id at de se pe il fil co AV **co**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
Connect	co	AWS_ENDPOINT_URL_CONNECT
Connect Contact Lens	co on ns	AWS_ENDPOINT_URL_CONNECT_CONTACT_LENS
ConnectCampaigns	co mp	AWS_ENDPOINT_URL_CONNECTCAMPAIGNS
ConnectCases	co se	AWS_ENDPOINT_URL_CONNECTCASES
ConnectParticipant	co rt	AWS_ENDPOINT_URL_CONNECTPARTICIPANT
ControlTower	co we	AWS_ENDPOINT_URL_CONTROLTOWER
Cost Optimization Hub	co m: hu	AWS_ENDPOINT_URL_COST_OPTIMIZATION_HUB

serviceId	Cl id at de se pe il fil co AV **ce**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
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
DataSync	da	AWS_ENDPOINT_URL_DATASYNC
DataZone	da	AWS_ENDPOINT_URL_DATAZONE
DAX	da	AWS_ENDPOINT_URL_DAX
Detective	de	AWS_ENDPOINT_URL_DETECTIVE
Device Farm	de rn	AWS_ENDPOINT_URL_DEVICE_FARM

serviceId	Cl id at de se pe il fil cc A\ **ce**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
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
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	di	AWS_ENDPOINT_URL_DRS
Directory Service	di _s	AWS_ENDPOINT_URL_DIRECTORY_SERVICE

serviceId	Cl id at de se pe il fil cc A\ **cc**	Variabile di ambiente **AWS_ENDPOINT_URL_<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 ne ct	AWS_ENDPOINT_URL_EC2_INSTANCE_CONNECT
ECR	ec	AWS_ENDPOINT_URL_ECR
ECR PUBLIC	ec c	AWS_ENDPOINT_URL_ECR_PUBLIC
ECS	ec	AWS_ENDPOINT_URL_ECS
EFS	ef	AWS_ENDPOINT_URL_EFS
EKS	ek	AWS_ENDPOINT_URL_EKS
EKS Auth	ek	AWS_ENDPOINT_URL_EKS_AUTH

serviceId	Cl id at de se pe il fil co AV **co**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
Elastic Inference	e: n:	AWS_ENDPOINT_URL_ELASTIC_INFERENCE
ElastiCache	e: he	AWS_ENDPOINT_URL_ELASTICACHE
Elastic Beanstalk	e: ea	AWS_ENDPOINT_URL_ELASTIC_BEANSTALK
Elastic Transcoder	e: ra	AWS_ENDPOINT_URL_ELASTIC_TRANSCODER
Elastic Load Balancing	e: oa c:	AWS_ENDPOINT_URL_ELASTIC_LOAD_BALANCING
Elastic Load Balancing v2	e: oa c:	AWS_ENDPOINT_URL_ELASTIC_LOAD_BALANCING_V2
EMR	er	AWS_ENDPOINT_URL_EMR
EMR containers	er ir	AWS_ENDPOINT_URL_EMR_CONTAINERS

serviceId	Cl id at de se pe il fil cc A\ **cc**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
EMR Serverless	er r:	AWS_ENDPOINT_URL_EMR_SERVERLESS
EntityResolution	er o:	AWS_ENDPOINT_URL_ENTITYRESOLUTION
Elasticsearch Service	e: a: ic	AWS_ENDPOINT_URL_ELASTICSEARCH_SERVICE
EventBridge	ev ge	AWS_ENDPOINT_URL_EVENTBRIDGE
Evidently	ev	AWS_ENDPOINT_URL_EVIDENTLY
finspace	f:	AWS_ENDPOINT_URL_FINSPACE
finspace data	f: da	AWS_ENDPOINT_URL_FINSPACE_DATA
Firehose	f:	AWS_ENDPOINT_URL_FIREHOSE
fis	f:	AWS_ENDPOINT_URL_FIS
FMS	fr	AWS_ENDPOINT_URL_FMS

serviceId	Cl id at de se pe il fil cc A\ **ce**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
forecast	fo	AWS_ENDPOINT_URL_FORECAST
forecastquery	fo ue	AWS_ENDPOINT_URL_FORECASTQUERY
FraudDetector	f: ct	AWS_ENDPOINT_URL_FRAUDDETECTOR
FreeTier	f:	AWS_ENDPOINT_URL_FREETIER
FSx	fs	AWS_ENDPOINT_URL_FSX
GameLift	ga	AWS_ENDPOINT_URL_GAMELIFT
Glacier	gl	AWS_ENDPOINT_URL_GLACIER
Global Accelerator	gl ce	AWS_ENDPOINT_URL_GLOBAL_ACCELERATOR
Glue	gl	AWS_ENDPOINT_URL_GLUE
grafana	gr	AWS_ENDPOINT_URL_GRAFANA
Greengrass	gr s	AWS_ENDPOINT_URL_GREENGRASS

serviceId	Cl id at de se pe il fil cc A\ **cc**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
GreengrassV2	g: sv	AWS_ENDPOINT_URL_GREENGRASSV2
GroundStation	g: t:	AWS_ENDPOINT_URL_GROUNDSTATION
GuardDuty	gu	AWS_ENDPOINT_URL_GUARDDUTY
Health	he	AWS_ENDPOINT_URL_HEALTH
HealthLake	he e	AWS_ENDPOINT_URL_HEALTHLAKE
Honeycode	ho	AWS_ENDPOINT_URL_HONEYCODE
IAM	ia	AWS_ENDPOINT_URL_IAM
identitystore	id to	AWS_ENDPOINT_URL_IDENTITYSTORE
imagebuilder	im de	AWS_ENDPOINT_URL_IMAGEBUILDER
ImportExport	im o:	AWS_ENDPOINT_URL_IMPORTEXPORT

serviceId	Cl id at de se pe il fil cc A\ **cc**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
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
IoT Data Plane	io p.	AWS_ENDPOINT_URL_IOT_DATA_PLANE
IoT Jobs Data Plane	io da e	AWS_ENDPOINT_URL_IOT_JOBS_DATA_PLANE
IoT 1Click Devices Service	io k_ _s	AWS_ENDPOINT_URL_IOT_1CLICK_DEVICES_ SERVICE

serviceId	Cl id at de se pe il fil cc A\ **co**	Variabile di ambiente **AWS_ENDPOINT_URL_<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
IoTFleetHub	ic ul	AWS_ENDPOINT_URL_IOTFLEETHUB
IoTFleetWise	ic is	AWS_ENDPOINT_URL_IOTFLEETWISE
IoTSecureTunneling	ic tu	AWS_ENDPOINT_URL_IOTSECURETUNNELING

serviceId	Cl id at de se pe il fil cc A\ **cc**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
IoTSiteWise	ic se	AWS_ENDPOINT_URL_IOTSITEWISE
IoTThingsGraph	ic g:	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
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

serviceId	Cl id at de se pe il fil cc A\ **cc**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
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_ARCHIVED_MEDIA
Kinesis Video Media	ki id a	AWS_ENDPOINT_URL_KINESIS_VIDEO_MEDIA
Kinesis Video Signaling	ki id a:	AWS_ENDPOINT_URL_KINESIS_VIDEO_SIGNALING
Kinesis Video WebRTC Storage	ki id to e	AWS_ENDPOINT_URL_KINESIS_VIDEO_WEBRTC_STORAGE

serviceId	Cl id at de se pe il fil co AV **co**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
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
Launch Wizard	la za	AWS_ENDPOINT_URL_LAUNCH_WIZARD
Lex Model Building Service	le _l _s	AWS_ENDPOINT_URL_LEX_MODEL_BUILDING_ SERVICE

serviceId	Cl id at de se pe il fil cc A\ **cc**	Variabile di ambiente **AWS_ENDPOINT_URL_<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	l: ar	AWS_ENDPOINT_URL_LICENSE_MANAGER
License Manager Linux Subscriptions	l: ar nu r:	AWS_ENDPOINT_URL_LICENSE_MANAGER_LIN UX_SUBSCRIPTIONS
License Manager User Subscriptions	l: ar e: ip	AWS_ENDPOINT_URL_LICENSE_MANAGER_USE R_SUBSCRIPTIONS
Lightsail	l:	AWS_ENDPOINT_URL_LIGHTSAIL

serviceId	Cl id at de se pe il fil cc A\ **cc**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
Location	lc	AWS_ENDPOINT_URL_LOCATION
CloudWatch Logs	c: h_	AWS_ENDPOINT_URL_CLOUDWATCH_LOGS
LookoutEquipment	lc u:	AWS_ENDPOINT_URL_LOOKOUTEQUIPMENT
LookoutMetrics	lc t:	AWS_ENDPOINT_URL_LOOKOUTMETRICS
LookoutVision	lc s:	AWS_ENDPOINT_URL_LOOKOUTVISION
m2	m2	AWS_ENDPOINT_URL_M2
Machine Learning	ma ea	AWS_ENDPOINT_URL_MACHINE_LEARNING
Macie2	ma	AWS_ENDPOINT_URL_MACIE2
ManagedBlockchain	ma oc	AWS_ENDPOINT_URL_MANAGEDBLOCKCHAIN
ManagedBlockchain Query	ma oc qu	AWS_ENDPOINT_URL_MANAGEDBLOCKCHAIN_QUERY

serviceId	Cl id at de se pe il fil cc A\ **ce**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
Marketplace Agreement	ma ce er	AWS_ENDPOINT_URL_MARKETPLACE_AGREEMENT
Marketplace Catalog	ma ce g	AWS_ENDPOINT_URL_MARKETPLACE_CATALOG
Marketplace Deployment	ma ce me	AWS_ENDPOINT_URL_MARKETPLACE_DEPLOYMENT
Marketplace Entitlement Service	ma ce er v:	AWS_ENDPOINT_URL_MARKETPLACE_ENTITLEMENT_SERVICE
Marketplace Commerce Analytics	ma ce ce ic	AWS_ENDPOINT_URL_MARKETPLACE_COMMERCE_ANALYTICS
MediaConnect	me ec	AWS_ENDPOINT_URL_MEDIACONNECT

serviceId	Cl id at de se pe il fil cc A\ **co**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
MediaConvert	me e:	AWS_ENDPOINT_URL_MEDIACONVERT
MediaLive	me	AWS_ENDPOINT_URL_MEDIALIVE
MediaPackage	me ag	AWS_ENDPOINT_URL_MEDIAPACKAGE
MediaPackage Vod	me ag	AWS_ENDPOINT_URL_MEDIAPACKAGE_VOD
MediaPackageV2	me ag	AWS_ENDPOINT_URL_MEDIAPACKAGEV2
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

serviceId	Cl id at de se pe il fil cc A\ **ce**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
MemoryDB	me	AWS_ENDPOINT_URL_MEMORYDB
Marketplace Metering	ma ce ng	AWS_ENDPOINT_URL_MARKETPLACE_METERING
Migration Hub	m: _I	AWS_ENDPOINT_URL_MIGRATION_HUB
mgn	mg	AWS_ENDPOINT_URL_MGN
Migration Hub Refactor Spaces	m: _I ct es	AWS_ENDPOINT_URL_MIGRATION_HUB_REFAC TOR_SPACES
MigrationHub Config	m: hu g	AWS_ENDPOINT_URL_MIGRATIONHUB_CONFIG
MigrationHubOrches trator	m: hu t:	AWS_ENDPOINT_URL_MIGRATIONHUBORCHESTRATOR

serviceId	Cl id at de se pe il fil co AV **co**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
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	mw	AWS_ENDPOINT_URL_MWAA
Neptune	ne	AWS_ENDPOINT_URL_NEPTUNE
Neptune Graph	ne ra	AWS_ENDPOINT_URL_NEPTUNE_GRAPH
neptunedata	ne ta	AWS_ENDPOINT_URL_NEPTUNEDATA
Network Firewall	ne i:	AWS_ENDPOINT_URL_NETWORK_FIREWALL
NetworkManager	ne na	AWS_ENDPOINT_URL_NETWORKMANAGER

serviceId	Cl id at de se pe il fil co A\ **co**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
NetworkMonitor	ne n:	AWS_ENDPOINT_URL_NETWORKMONITOR
nimble	n:	AWS_ENDPOINT_URL_NIMBLE
OAM	oa	AWS_ENDPOINT_URL_OAM
Omics	or	AWS_ENDPOINT_URL_OMICS
OpenSearch	op h	AWS_ENDPOINT_URL_OPENSEARCH
OpenSearchServerless	op hs ss	AWS_ENDPOINT_URL_OPENSEARCHSERVERLESS
OpsWorks	op	AWS_ENDPOINT_URL_OPSWORKS
OpsWorksCM	op m	AWS_ENDPOINT_URL_OPSWORKSCM
Organizations	or io	AWS_ENDPOINT_URL_ORGANIZATIONS
OSIS	os	AWS_ENDPOINT_URL_OSIS
Outposts	ou	AWS_ENDPOINT_URL_OUTPOSTS

serviceId	Cl id at de se pe il fil cc A\ **ce**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
p8data	p8	AWS_ENDPOINT_URL_P8DATA
p8data	p8	AWS_ENDPOINT_URL_P8DATA
Panorama	pa	AWS_ENDPOINT_URL_PANORAMA
Payment Cryptography	pa ry hy	AWS_ENDPOINT_URL_PAYMENT_CRYPTOGRAPHY
Payment Cryptography Data	pa ry hy	AWS_ENDPOINT_URL_PAYMENT_CRYPTOGRAPHY_DATA
Pca Connector Ad	pc ct	AWS_ENDPOINT_URL_PCA_CONNECTOR_AD
Personalize	pe ze	AWS_ENDPOINT_URL_PERSONALIZE
Personalize Events	pe ze	AWS_ENDPOINT_URL_PERSONALIZE_EVENTS

serviceId	Cl id at de se pe il fil cc A\ **ce**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
Personalize Runtime	pe ze e	AWS_ENDPOINT_URL_PERSONALIZE_RUNTIME
PI	p:	AWS_ENDPOINT_URL_PI
Pinpoint	p:	AWS_ENDPOINT_URL_PINPOINT
Pinpoint Email	p: er	AWS_ENDPOINT_URL_PINPOINT_EMAIL
Pinpoint SMS Voice	p: sr	AWS_ENDPOINT_URL_PINPOINT_SMS_VOICE
Pinpoint SMS Voice V2	p: sr _\	AWS_ENDPOINT_URL_PINPOINT_SMS_VOICE_V2
Pipes	p:	AWS_ENDPOINT_URL_PIPES
Polly	po	AWS_ENDPOINT_URL_POLLY
Pricing	p:	AWS_ENDPOINT_URL_PRICING
PrivateNetworks	p: tv	AWS_ENDPOINT_URL_PRIVATENETWORKS

serviceId	Cl id at de se pe il fil cc A\ **ce**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
Proton	p:	AWS_ENDPOINT_URL_PROTON
QBusiness	qb	AWS_ENDPOINT_URL_QBUSINESS
QConnect	qc	AWS_ENDPOINT_URL_QCONNECT
QLDB	q:	AWS_ENDPOINT_URL_QLDB
QLDB Session	q: ic	AWS_ENDPOINT_URL_QLDB_SESSION
QuickSight	qu t	AWS_ENDPOINT_URL_QUICKSIGHT
RAM	ra	AWS_ENDPOINT_URL_RAM
rbin	rb	AWS_ENDPOINT_URL_RBIN
RDS	rc	AWS_ENDPOINT_URL_RDS
RDS Data	rc	AWS_ENDPOINT_URL_RDS_DATA
Redshift	re	AWS_ENDPOINT_URL_REDSHIFT
Redshift Data	re da	AWS_ENDPOINT_URL_REDSHIFT_DATA

serviceId	Cl id at de se pe il fil cc A\ **ce**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
Redshift Serverless	re se s	AWS_ENDPOINT_URL_REDSHIFT_SERVERLESS
Rekognition	re on	AWS_ENDPOINT_URL_REKOGNITION
repostspace	re ce	AWS_ENDPOINT_URL_REPOSTSPACE
resiliencehub	re eh	AWS_ENDPOINT_URL_RESILIENCEHUB
Resource Explorer 2	re ex 2	AWS_ENDPOINT_URL_RESOURCE_EXPLORER_2
Resource Groups	re g:	AWS_ENDPOINT_URL_RESOURCE_GROUPS
Resource Groups Tagging API	re g: g(	AWS_ENDPOINT_URL_RESOURCE_GROUPS_TAG GING_API

serviceId	Cl id at de se pe il fil co A\ **co**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
RoboMaker	rc	AWS_ENDPOINT_URL_ROBOMAKER
RolesAnywhere	rc he	AWS_ENDPOINT_URL_ROLESANYWHERE
Route 53	rc	AWS_ENDPOINT_URL_ROUTE_53
Route53 Recovery Cluster	rc ec lu	AWS_ENDPOINT_URL_ROUTE53_RECOVERY_CLUSTER
Route53 Recovery Control Config	rc ec or nt	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 do	AWS_ENDPOINT_URL_ROUTE_53_DOMAINS
Route53Resolver	rc so	AWS_ENDPOINT_URL_ROUTE53RESOLVER

serviceId	Cl id at de se pe il fil cc A\ **ce**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
RUM	ru	AWS_ENDPOINT_URL_RUM
S3	s3	AWS_ENDPOINT_URL_S3
S3 Control	s3 l	AWS_ENDPOINT_URL_S3_CONTROL
S3Outposts	s3 s	AWS_ENDPOINT_URL_S3OUTPOSTS
SageMaker	sa	AWS_ENDPOINT_URL_SAGEMAKER
SageMaker A2I Runtime	sa _a ir	AWS_ENDPOINT_URL_SAGEMAKER_A2I_RUNTIME
Sagemaker Edge	sa _e	AWS_ENDPOINT_URL_SAGEMAKER_EDGE
SageMaker FeatureSt ore Runtime	sa _f to ir	AWS_ENDPOINT_URL_SAGEMAKER_FEATUREST ORE_RUNTIME

serviceId	Cl id at de se pe il fil cc A\ **ce**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
SageMaker Geospatial	sa _g ai	AWS_ENDPOINT_URL_SAGEMAKER_GEOSPATIAL
SageMaker Metrics	sa _r	AWS_ENDPOINT_URL_SAGEMAKER_METRICS
SageMaker Runtime	sa _r	AWS_ENDPOINT_URL_SAGEMAKER_RUNTIME
savingsplans	sa ar	AWS_ENDPOINT_URL_SAVINGSPLANS
Scheduler	sc	AWS_ENDPOINT_URL_SCHEDULER
schemas	sc	AWS_ENDPOINT_URL_SCHEMAS
SimpleDB	sd	AWS_ENDPOINT_URL_SIMPLEDB
Secrets Manager	se ar	AWS_ENDPOINT_URL_SECRETS_MANAGER
SecurityHub	se ub	AWS_ENDPOINT_URL_SECURITYHUB

serviceId	Cl id at de se pe il fil cc A\ **ce**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
SecurityLake	se al	AWS_ENDPOINT_URL_SECURITYLAKE
ServerlessApplicationRepository	se sa ic to	AWS_ENDPOINT_URL_SERVERLESSAPPLICATIONREPOSITORY
Service Quotas	se uc	AWS_ENDPOINT_URL_SERVICE_QUOTAS
Service Catalog	se at	AWS_ENDPOINT_URL_SERVICE_CATALOG
Service Catalog AppRegistry	se at p:	AWS_ENDPOINT_URL_SERVICE_CATALOG_APPREGISTRY
ServiceDiscovery	se sc	AWS_ENDPOINT_URL_SERVICEDISCOVERY
SES	se	AWS_ENDPOINT_URL_SES
SESv2	se	AWS_ENDPOINT_URL_SESV2

serviceId	Cl id at de se pe il fil cc A\ **ce**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
Shield	sh	AWS_ENDPOINT_URL_SHIELD
signer	s:	AWS_ENDPOINT_URL_SIGNER
SimSpaceWeaver	s: ea	AWS_ENDPOINT_URL_SIMSPACEWEAVER
SMS	sn	AWS_ENDPOINT_URL_SMS
Snow Device Management	sn ce me	AWS_ENDPOINT_URL_SNOW_DEVICE_MANAGEMENT
Snowball	sn	AWS_ENDPOINT_URL_SNOWBALL
SNS	sn	AWS_ENDPOINT_URL_SNS
SQS	sq	AWS_ENDPOINT_URL_SQS
SSM	ss	AWS_ENDPOINT_URL_SSM
SSM Contacts	ss ct	AWS_ENDPOINT_URL_SSM_CONTACTS
SSM Incidents	ss en	AWS_ENDPOINT_URL_SSM_INCIDENTS

serviceId	Cl id at de se pe il fil co AV **co**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
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
STS	st	AWS_ENDPOINT_URL_STS
SupplyChain	su in	AWS_ENDPOINT_URL_SUPPLYCHAIN
Support	su	AWS_ENDPOINT_URL_SUPPORT
Support App	su pp	AWS_ENDPOINT_URL_SUPPORT_APP
SWF	sw	AWS_ENDPOINT_URL_SWF

serviceId	Cl id at de se pe il fil cc A\ **cc**	Variabile di ambiente **AWS_ENDPOINT_URL_<SERVICE>**
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
tnb	tr	AWS_ENDPOINT_URL_TNB
Transcribe	ti e	AWS_ENDPOINT_URL_TRANSCRIBE
Transfer	ti	AWS_ENDPOINT_URL_TRANSFER
Translate	ti	AWS_ENDPOINT_URL_TRANSLATE
TrustedAdvisor	ti v	AWS_ENDPOINT_URL_TRUSTEDADVISOR

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

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

Impostazioni predefinite di configurazione intelligente

> **Note**
>
> Per informazioni sulla comprensione del layout delle pagine delle impostazioni o sull'interpretazione della tabella Support by AWS SDKs and tools riportata di seguito, vedereInformazioni sulle pagine delle impostazioni di questa guida.

Con la funzione smart configuration defaults, è AWS SDKs possibile 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, è possibile 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: le nuove versioni principali di SDKs avranno come impostazione predefinita. `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 individuare 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, il che potrebbe introdurre latenza. Se la latenza di avvio è fondamentale per l'applicazione, si consiglia invece di scegliere una `defaults_mode` esplicita.

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 regionali.
- `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`

Parametro	Modalità **standard**	Modalità **in-region**	Modalità **cross-region**	Modalità **mobile**
`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).

Support by AWS SDKs and tools

Di seguito sono SDKs supportate 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 per Java and the. AWS SDK per Kotlin

SDK	Supportata	Note o ulteriori informazioni
AWS CLI v2	No	
SDK per C++	Sì	Parametri non ottimizzati:`stsRegionalEndpoints` ,,`s3UsEast1`

SDK	Supportata	Note o ulteriori informazioni
		`RegionalEndpoints` . `tlsNegotiationTime outInMillis`
SDK per Go V2 (1.x)	Sì	Parametri non ottimizza ti:`retryMode` ,,. `stsRegion alEndpoints` `s3UsEast1 RegionalEndpoints`
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 ottimizza ti:`stsRegionalEndpoin ts` ,,`s3UsEast1 RegionalEndpoints` . `tlsNegotiationTime outInMillis` `connectTi meoutInMillis` viene chiamato`connectio nTimeout` .
SDK per 2.x JavaScript	No	
SDK per Kotlin	No	
SDK per.NET 4.x	Sì	Parametri non ottimizza ti:`connectTimeoutInMi llis` ,. `tlsNegoti ationTimeoutInMill is`

SDK	Supportata	Note o ulteriori informazioni
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	
SDK per Swift	No	
Strumenti per V5 PowerShell	Sì	Parametri non ottimizza ti:`connectTimeoutInMillis` ,`tlsNegotiationTimeoutInMillis` .
Strumenti per PowerShell V4	Sì	Parametri non ottimizza ti:`connectTimeoutInMillis` ,`tlsNegotiationTimeoutInMillis` .

AWS Librerie Common Runtime (CRT)

Le librerie AWS Common Runtime (CRT) sono una libreria di base di. SDKs 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 da tutti e SDKs forniscono 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,, HMAC), firmatari, AES SHA256 SHA256
- `awslabs/aws-c-common`: Strutture dati di base, tipi threading/synchronization primitivi, 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 Event Stream (headers, prelude, payload, crc/trailer), implementazione di chiamate di procedura remota (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 a larghezza di banda elevata EC2
- `awslabs/aws-c-sdkutils`: Una libreria di utilità per l'analisi e la gestione dei profili AWS
- `awslabs/aws-checksums`: accelerata dall'hardware multipiattaforma e con possibilità di ricorrere a implementazioni software CRC32c efficienti CRC32
- `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 per tutti SDKs tranne Go e Rust.

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.
Diagramma delle dipendenze CRT che mostra come le singole librerie CRT interagiscono tra loro.

AWS SDKs e politica di manutenzione degli strumenti

Panoramica di

Questo documento delinea la politica di manutenzione per i kit e gli strumenti di sviluppo AWS software (SDKs), inclusi dispositivi mobili e IoT SDKs, e le relative dipendenze sottostanti. AWS fornisce regolarmente a AWS SDKs and Tools aggiornamenti che possono contenere supporto per nuove funzionalità, miglioramenti AWS APIs, correzioni di bug, patch di sicurezza o aggiornamenti della documentazione nuovi o aggiornati. 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 versioni principali SDKs e di Tools è costituito da 5 fasi, descritte di seguito.

- Anteprima per sviluppatori (Fase 0): durante questa fase, non SDKs sono supportate, non devono essere utilizzate in ambienti di produzione e sono pensate 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, SDKs 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.
- End-of-Support (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.
Tempistiche della politica di manutenzione

Ciclo di vita delle dipendenze

La maggior parte AWS SDKs 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 deprecazione vengono aggiunti alla documentazione SDK SDKs, che delineano il percorso e il collegamento alla documentazione dell'SDK. end-of-support

Per visualizzare l'elenco delle versioni principali disponibili di AWS SDKs and Tools e la relativa fase del ciclo di manutenzione, consulta. Ciclo di vita delle versioni

AWS SDKs ciclo di vita delle versioni e degli strumenti

La tabella seguente mostra l'elenco delle versioni principali del AWS Software Development Kit (SDK) disponibili e la loro collocazione nel ciclo di vita della manutenzione con le relative tempistiche. Per informazioni dettagliate sul ciclo di vita delle versioni principali di AWS SDKs and Tools e sulle relative dipendenze sottostanti, vedere. Politica di manutenzione

SDK	Versione principale	Fase attuale	Data di disponibilità generale	Note
AWS CLI	1.x	Annuncio di manutenzione	2/09/2013	Vedi l'annuncio per dettagli e date
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	Fine del supporto	19/11/2015	
SDK per Java	1.x	Fine del supporto	25/03/2010	
SDK per Java	2.x	Disponibilità generale	20/11/2018	
SDK per JavaScript	1.x	Fine del supporto	6/05/2013	
SDK per JavaScript	2.x	Fine del supporto	19/06/2014	

SDK	Versione principale	Fase attuale	Data di disponibilità generale	Note
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.NET	4.x	Disponibilità generale	28/04/2025	
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	Disponibilità generale	17/09/2024	
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	
Utensili per PowerShell	5.x	Disponibilità generale	23/06/2025	

Cerchi un SDK o uno strumento non menzionato? Crittografia SDKs SDKs, dispositivo IoT e dispositivi mobili SDKs, ad esempio, non sono inclusi in questa guida. Per trovare la documentazione su questi altri strumenti, consulta Tools to Build on AWS.

Cronologia dei documenti AWS SDKs e guida di riferimento agli strumenti

La tabella seguente descrive importanti aggiunte e aggiornamenti alla AWS SDKs and Tools Reference Guide. Per ricevere notifiche sugli aggiornamenti della documentazione, puoi sottoscrivere il feed RSS.

Modifica	Descrizione	Data
Aggiungere una nuova impostazione S3 Express One Zone	Aggiunta una nuova impostazione S3 Express One Zone per disabilitare l'autenticazione della sessione.	13 ottobre 2025
Aggiungere un nuovo albero decisionale di autenticazione	Aggiunta di un nuovo albero decisionale per facilitare le decisioni di autenticazione tra le opzioni.	23 settembre 2025
Aggiunta di una nuova funzionalità dello schema di autenticazione	Aggiunta di una nuova funzionalità dello schema di autenticazione. Aggiornamenti agli endpoint AWS STS regionali.	18 agosto 2025
Aggiunta una nuova versione di Tools for PowerShell	Aggiunta della versione più recente di Tools per PowerShell il supporto a tutte le impostazioni di riferimento Compatibilità con AWS SDKs le tabelle. Aggiunta la funzione di iniezione del prefisso Host.	23 giugno 2025
Aggiornamenti del titolo della pagina	Altri titoli, titoli di tabelle, abstract e aggiornamenti SEO.	5 marzo 2025

Aggiornamenti del titolo della pagina	Aggiornamento dei contenuti per utilizzare titoli più descritti vi.	24 febbraio 2025
Aggiungere Swift SDK al riferimento Impostazioni	Aggiungere il supporto Swift SDK a tutte le impostazioni di riferimento Compatibilità con le tabelle. AWS SDKs	17 settembre 2024
Proprietà di sistema SDK for Java 1.x	Aggiungi dettagli sulle impostazioni di configurazione del sistema JVM supportate dalla versione 1.x. AWS SDK per 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 IMDSv1 fallback.	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 per Rust e. AWS Strumenti per PowerShell	10 dicembre 2023
Aggiornamenti di autentica zione	Aggiornamenti ai metodi SDKs e agli strumenti di autentica zione supportati.	1 luglio 2023
Aggiornamenti delle best practice di IAM	Guida aggiornata per l'allinea mento alle best practice IAM. Per ulteriori informazioni, consulta Best practice per la sicurezza in IAM.	27 febbraio 2023
Aggiornamenti SSO	Aggiornamenti alle credenzia li SSO per la nuova configura zione del token SSO.	19 novembre 2022
Aggiornamenti delle impostazi oni	Aggiornamenti alla tabella di supporto per la configura zione generale e per i punti di accesso multiregionali di Amazon S3.	17 novembre 2022
Aggiornamenti delle impostazi oni	Aggiornamenti alla chiarezza del client IMDS e delle credenziali IMDS. Aggiornam enti 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

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.