



Guide de référence

AWS SDK et outils



AWS SDK et outils: Guide de référence

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

Les marques et la présentation commerciale d'Amazon ne peuvent être utilisées en relation avec un produit ou un service qui n'est pas d'Amazon, d'une manière susceptible de créer une confusion parmi les clients, ou d'une manière qui dénigre ou discrédite Amazon. Toutes les autres marques commerciales qui ne sont pas la propriété d'Amazon appartiennent à leurs propriétaires respectifs, qui peuvent ou non être affiliés ou connectés à Amazon, ou sponsorisés par Amazon.

Table of Contents

AWS Guide de référence des SDK et des outils	1
Ressources pour les développeurs	2
Notification de télémétrie du kit d'outils	3
Configuration	4
Partage config et credentials fichiers	5
Profils	5
Format du fichier de configuration	7
Format du fichier d'informations d'identification	10
Emplacement des fichiers partagés	11
Résolution du répertoire personnel	11
Modifier l'emplacement par défaut de ces fichiers	12
Variables d'environnement	13
Comment définir les variables d'environnement	13
Configuration de variables d'environnement sans serveur	15
Propriétés du système JVM	15
Comment définir les propriétés du système JVM	16
Authentification et accès	18
ID de constructeur AWS	20
Authentification IAM Identity Center	20
Configuration de l'accès programmatique à l'aide d'IAM Identity Center	21
Comprendre l'authentification IAM Identity Center	24
Rôles Anywhere IAM	28
Étape 1 : configurer les rôles IAM n'importe où	29
Étape 2 : Utiliser les rôles IAM n'importe où	29
Assumer un rôle	30
Assumez un rôle IAM	31
Fédérez avec l'identité Web ou OpenID Connect	32
AWS clés d'accès	34
Utiliser des identifiants à court terme	34
Utilisez des identifiants à long terme	34
Identifiants à court terme	36
Des références à long terme	37
Rôles IAM pour les instances Amazon EC2	41
Créer un rôle IAM	41

Lancez une instance Amazon EC2 et spécifiez votre rôle IAM	41
Connectez-vous à l'instance EC2	42
Exécutez l'exemple d'application sur l'instance EC2	42
Référence des paramètres	44
Création de clients de service	44
Priorité des paramètres	44
Configliste des paramètres de fichier	46
Credentialsliste des paramètres de fichier	49
Liste des variables d'environnement	50
Liste des propriétés du système JVM	54
Fournisseurs d'identifiants standardisés	56
Chaîne de fournisseurs d'identifiants	57
AWS clés d'accès	58
Assumer le rôle de fournisseur	62
Fournisseur de conteneurs	68
Fournisseur de centre d'identité IAM	71
fournisseur IMDS	78
Fournisseur de processus	83
Fonctionnalités standardisées	87
ID d'application	88
Métadonnées d'instance Amazon EC2.	90
Points d'accès Amazon S3	92
Amazon S3 Multi-Region Access Points	95
Région AWS	97
AWS STS Points de terminaison régionalisés	100
Points de terminaison à double pile et FIPS	102
Découverte du points de terminaison	105
Configuration générale	107
Client IMDS	110
Comportement de nouvelle tentative	113
Compression des demandes	119
Points de terminaison spécifiques au service	121
Paramètres de configuration intelligents par défaut	170
Runtime commun	176
Dépendances CRT	177
Politique de maintenance	178

Présentation	178
Gestion des versions	178
Cycle de vie des versions majeures du SDK	178
Cycle de vie des dépendances	179
Méthodes de communication	180
Matrice de support des versions	181
Historique de la documentation	184
Glossaire AWS	187
.....	clxxxviii

AWS Guide de référence des SDK et des outils

De nombreux SDK et outils partagent certaines fonctionnalités communes, soit par le biais de spécifications de conception partagées, soit par le biais d'une bibliothèque partagée.

Ce guide contient des informations concernant :

- [Configuration](#)— Comment utiliser le partage, config les credentials fichiers ou les variables d'environnement pour configurer vos AWS SDK et vos outils.
- [Authentification et accès](#)— Déterminez comment votre code ou outil s'authentifie AWS lorsque vous développez avec Services AWS.
- [Référence des paramètres](#)— Référence pour tous les paramètres standardisés disponibles pour l'authentification et la configuration.
- [AWSbibliothèques CRT \(Common Runtime\)](#)— Vue d'ensemble des bibliothèques CRT (AWS Common Runtime) partagées disponibles pour presque tous les SDK.
- [AWS Politique de maintenance des SDK et des outils](#) couvre la politique de maintenance et le contrôle des versions pour les kits de développement AWS logiciel (SDK) et les outils, y compris les SDK mobiles et Internet des objets (IoT), ainsi que leurs dépendances sous-jacentes.

Ce guide de référence sur AWS les SDK et les outils est destiné à constituer une base d'informations applicable à plusieurs SDK et outils. Le guide spécifique au SDK ou à l'outil que vous utilisez doit être utilisé en plus des informations présentées ici. Vous trouverez ci-dessous le SDK et les outils qui contiennent des sections pertinentes dans ce guide :

Si vous utilisez :	Les sections pertinentes de ce guide pour vous sont les suivantes :
<ul style="list-style-type: none"> • N'importe quel SDK ou outil 	AWS Politique de maintenance des SDK et des outils
<ul style="list-style-type: none"> • AWS Cloud Development Kit (AWS CDK) Manuel du développeur • AWS Serverless Application Model Manuel du développeur • AWS Toolkit for Eclipse Guide de l'utilisateur 	Configuration Authentification et accès AWS Politique de maintenance des SDK et des outils

Si vous utilisez :	Les sections pertinentes de ce guide pour vous sont les suivantes :
<ul style="list-style-type: none">• AWS Toolkit for JetBrains Guide de l'utilisateur• AWS Toolkit for Visual Studio Guide de l'utilisateur• AWS Toolkit for Visual Studio Code Guide de l'utilisateur	
<ul style="list-style-type: none">• AWS Command Line Interface Guide de l'utilisateur• AWS SDK for C++ Manuel du développeur• AWS SDK for Go Manuel du développeur• AWS SDK for Java Manuel du développeur• AWS SDK for JavaScript Manuel du développeur• Kit AWS SDK pour Kotlin• AWS SDK for .NET Manuel du développeur• AWS SDK for PHP Manuel du développeur• AWS SDK pour Python (Boto3) Getting Started• AWS SDK for Ruby Manuel du développeur• Kit AWS SDK pour Rust• Kit AWS SDK pour Swift• AWS Tools for Windows PowerShell Guide de l'utilisateur	<ul style="list-style-type: none">• Configuration• Authentification et accès• Référence des paramètres• AWSbibliothèques CRT (Common Runtime)• AWS Politique de maintenance des SDK et des outils• AWS Matrice de prise en charge des versions des SDK et des outils

Ressources pour les développeurs

Pour un aperçu des outils qui peuvent vous aider à développer des applications AWS, consultez la section [Outils sur lesquels vous pouvez vous appuyer AWS](#). Pour plus d'informations sur le support, consultez le [centre de AWS connaissances](#).

Amazon Q Developer est un assistant conversationnel génératif alimenté par l'IA qui peut vous aider à comprendre, créer, étendre et exploiter des applications. AWS Pour accélérer votre développement AWS, le modèle sur lequel repose Amazon Q est complété par AWS du contenu de haute qualité afin de produire des réponses plus complètes, exploitables et référencées. Pour plus d'informations, consultez [Qu'est-ce qu'Amazon Q Developer ?](#) dans le guide de l'utilisateur Amazon Q Developer.

Notification de télémétrie du kit d'outils

AWS Les boîtes à outils de l'environnement de développement intégré (IDE) sont des plugins et des extensions qui permettent d'accéder aux AWS services depuis votre IDE. Pour des informations détaillées sur chacun des kits d'outils IDE, consultez les guides de l'utilisateur du kit d'outils dans le tableau précédent.

AWS Les kits d'outils IDE peuvent collecter et stocker des données de télémétrie côté client pour éclairer les décisions concernant les futures versions du Toolkit. AWS Les données collectées quantifient votre utilisation de la AWS boîte à outils.

Pour en savoir plus sur les données de télémétrie collectées dans tous les kits d'outils AWS IDE, consultez le document [CommonDefinitions.json dans le référentiel Github](#). `aws-toolkit-common`

Pour des informations détaillées sur les données de télémétrie collectées par chacun des kits d'outils AWS IDE, consultez les documents de ressources dans les référentiels Github des kits d' AWS outils suivants :

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

Certains AWS services accessibles dans les boîtes à AWS outils peuvent collecter des données de télémétrie supplémentaires côté client. Pour obtenir des informations détaillées sur le type de données collectées par chaque AWS service individuel, consultez la rubrique [AWS Documentation](#) du service spécifique qui vous intéresse.

Configuration

Avec AWS les SDK et autres outils AWS de développement, tels que le AWS Command Line Interface (AWS CLI), vous pouvez interagir avec les API AWS de service. Avant d'essayer, vous devez toutefois configurer le SDK ou l'outil avec les informations dont il a besoin pour effectuer l'opération demandée.

Ces informations incluent les éléments suivants :

- Informations d'identification qui identifient la personne qui appelle l'API. Les informations d'identification sont utilisées pour chiffrer la demande adressée aux AWS serveurs. À l'aide de ces informations, vous AWS confirmez votre identité et pouvez récupérer les politiques d'autorisation qui y sont associées. Il peut ensuite déterminer les actions que vous êtes autorisé à effectuer.
- Autres détails de configuration que vous utilisez pour indiquer au AWS CLI SDK comment traiter la demande, où envoyer la demande (à quel point de terminaison de AWS service) et comment interpréter ou afficher la réponse.

Chaque SDK ou outil prend en charge plusieurs sources que vous pouvez utiliser pour fournir les informations d'identification et de configuration requises. Certaines sources sont propres au SDK ou à l'outil, et vous devez consulter la documentation de cet outil ou de ce SDK pour savoir comment utiliser cette méthode.

Cependant, la plupart des AWS SDK et des outils prennent en charge les paramètres communs provenant de deux sources principales (au-delà du code lui-même) :

- [Fichiers de AWS configuration et d'identification](#) partagés : les `credentials` fichiers `config` et d'identification partagés constituent le moyen le plus courant de spécifier l'authentification et la configuration d'un AWS SDK ou d'un outil. Utilisez ces fichiers pour enregistrer les paramètres que vos outils et applications peuvent utiliser. Les paramètres du `config` et `credentials` des fichiers sont associés à un profil spécifique. Avec plusieurs profils, vous pouvez créer différentes configurations de paramètres à appliquer dans différents scénarios. Lorsque vous utilisez un AWS outil pour appeler une commande ou que vous utilisez un SDK pour appeler une AWS API, vous pouvez spécifier le profil, et donc les paramètres de configuration, à utiliser pour cette action. L'un des profils est désigné comme `default` profil et est utilisé automatiquement lorsque vous ne spécifiez pas explicitement le profil à utiliser. Les paramètres que vous pouvez enregistrer dans ces fichiers sont décrits dans ce guide de référence.

- [Variables d'environnement](#) — Certains paramètres peuvent également être stockés dans les variables d'environnement de votre système d'exploitation. Bien que vous ne puissiez avoir qu'un seul ensemble de variables d'environnement en vigueur à la fois, elles sont facilement modifiées de manière dynamique au fur et à mesure que votre programme s'exécute et que vos exigences changent.

Sujets supplémentaires dans cette section

- [Partage config et credentials fichiers](#)
- [Emplacement du partage config et des credentials fichiers](#)
- [Support des variables d'environnement](#)
- [Support des propriétés du système JVM](#)

Partage **config** et **credentials** fichiers

Les `credentials` fichiers partagés AWS `config` contiennent un ensemble de profils. Un profil est un ensemble de paramètres de configuration, par paires clé-valeur, utilisé par le AWS Command Line Interface (AWS CLI), les AWS SDK et d'autres outils. Les valeurs de configuration sont associées à un profil afin de configurer certains aspects du SDK ou de l'outil lorsque ce profil est utilisé. Ces fichiers sont « partagés » dans la mesure où les valeurs s'appliquent à toutes les applications, processus ou SDK de l'environnement local d'un utilisateur.

Les fichiers partagés `config` et `credentials` les fichiers sont des fichiers en texte brut contenant uniquement des caractères ASCII (encodés en UTF-8). Ils prennent la forme de ce que l'on appelle généralement des [fichiers INI](#).

Profils

Les paramètres du partage `config` et `credentials` des fichiers sont associés à un profil spécifique. Plusieurs profils peuvent être définis dans le fichier afin de créer différentes configurations de paramètres à appliquer dans différents environnements de développement.

Le `[default]` profil contient les valeurs utilisées par un SDK ou une opération d'outil si aucun profil nommé spécifique n'est spécifié. Vous pouvez également créer des profils distincts auxquels vous pouvez explicitement faire référence par leur nom. Chaque profil peut utiliser des paramètres et des valeurs différents selon les besoins de votre application et de votre scénario.

Note

[default] est simplement un profil anonyme. Ce profil est nommé default car il s'agit du profil par défaut utilisé par le SDK si l'utilisateur ne spécifie aucun profil. Il ne fournit pas de valeurs par défaut héritées aux autres profils. Si vous définissez un élément dans le [default] profil et que vous ne le définissez pas dans un profil nommé, la valeur n'est pas définie lorsque vous utilisez le profil nommé.

Définissez un profil nommé

Le [default] profil et plusieurs profils nommés peuvent exister dans le même fichier. Utilisez le paramètre suivant pour sélectionner les paramètres du profil utilisés par votre SDK ou votre outil lors de l'exécution de votre code. Les profils peuvent également être sélectionnés dans le code ou par commande lorsque vous travaillez avec le AWS CLI.

Configurez cette fonctionnalité en définissant l'une des options suivantes :

AWS_PROFILE- variable d'environnement

Lorsque cette variable d'environnement est définie sur un profil nommé ou « par défaut », tous les codes et AWS CLI commandes du SDK utilisent les paramètres de ce profil.

Exemple Linux/macOS de définition de variables d'environnement via la ligne de commande :

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

Exemple Windows de définition de variables d'environnement via la ligne de commande :

```
setx AWS_PROFILE "my_default_profile_name"
```

aws.profile- Propriété du système JVM

[Pour le SDK pour Kotlin sur la JVM et le SDK pour Java 2.x, vous pouvez définir la propriété du système. aws.profile](#) Lorsque le SDK crée un client de service, il utilise les paramètres du profil nommé, sauf si le paramètre est remplacé dans le code. Le SDK pour Java 1.x ne prend pas en charge cette propriété système.

Format du fichier de configuration

Le config fichier est organisé en sections. Une section est un ensemble nommé de paramètres qui continue jusqu'à ce qu'une autre ligne de définition de section soit rencontrée.

Le config fichier est un fichier en texte brut qui utilise le format suivant :

- Toutes les entrées d'une section prennent la forme générale `setting-name=value`.
- Les lignes peuvent être commentées en commençant par un hashtag (#).

Types de sections

Une définition de section est une ligne qui donne un nom à un ensemble de paramètres. Les lignes de définition de section commencent et se terminent par des crochets ([]). À l'intérieur des crochets, il y a un identifiant de type de section et un nom personnalisé pour la section. Vous pouvez utiliser des lettres, des chiffres, des traits d'union (-) et des traits de soulignement (_), mais pas d'espaces.

Type de section : **default**

Exemple de ligne de définition de section : `[default]`

`[default]` est le seul profil qui ne nécessite pas l'identifiant de profile section.

L'exemple suivant montre un config fichier de base avec un `[default]` profil. Il définit le [region](#) réglage. Tous les paramètres qui suivent cette ligne, jusqu'à ce qu'une autre définition de section soit trouvée, font partie de ce profil.

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

Type de section : **profile**

Exemple de ligne de définition de section : `[profile dev]`

La ligne de définition de profile section est un groupe de configuration nommé que vous pouvez appliquer à différents scénarios de développement. Pour mieux comprendre les profils nommés, consultez la section précédente sur les profils.

L'exemple suivant montre un config fichier avec une ligne de définition de `profile` section et un profil nommé appelé `foo`. Tous les paramètres qui suivent cette ligne, jusqu'à ce qu'une autre définition de section soit trouvée, font partie de ce profil nommé.

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

Certains paramètres possèdent leur propre groupe imbriqué de sous-paramètres, tels que le `s3` paramètre et les sous-paramètres de l'exemple suivant. Associez les sous-paramètres au groupe en les indentant d'un ou de plusieurs espaces.

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

Type de section : **sso-session**

Exemple de ligne de définition de section : `[sso-session my-sso]`

La ligne de définition de `sso-session` section nomme un groupe de paramètres que vous utilisez pour configurer un profil afin de résoudre les AWS informations d'identification utilisées AWS IAM Identity Center. Pour plus d'informations sur la configuration de l'authentification unique, consultez [Authentification IAM Identity Center](#). Un profil est lié à une `sso-session` section par une paire clé-valeur où `sso-session` est la clé et le nom de votre `sso-session` section est la valeur, par exemple. `sso-session = <name-of-sso-session-section>`

L'exemple suivant configure un profil qui obtiendra des informations d' AWS identification à court terme pour le rôle IAM « `SampleRole` » dans le compte « `111122223333` » à l'aide d'un jeton du « `my-sso` ». La section « `my-sso` » est référencée dans la `sso-session` `profile` section par son nom à l'aide de la `sso-session` clé.

```
[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
```

Type de section : **services**

Exemple de ligne de définition de section : [services *dev*]

Note

Cette `services` section prend en charge les personnalisations des points de terminaison spécifiques aux services et n'est disponible que dans les SDK et les outils qui incluent cette fonctionnalité. Pour savoir si cette fonctionnalité est disponible pour votre SDK, consultez la section relative aux points de terminaison [Compatibilité avec les AWS SDK](#) spécifiques au service.

La ligne de définition de `services` section nomme un groupe de paramètres qui configurent les points de terminaison personnalisés pour les Service AWS demandes. Un profil est lié à une `services` section par une paire clé-valeur où `services` est la clé et le nom de votre `services` section est la valeur, par exemple. `services = <name-of-services-section>`

La `services` section est ensuite séparée en sous-sections par des `<SERVICE> =` lignes, où se `<SERVICE>` trouve la clé d' Service AWS identification. L' Service AWS identifiant est basé sur le modèle d'API `serviceId` en remplaçant tous les espaces par des traits de soulignement et en minuscules toutes les lettres. Pour obtenir la liste de toutes les clés d'identification de service à utiliser dans `services` cette section, consultez [Identifiants pour les points de terminaison spécifiques au service](#). La clé d'identification du service est suivie de paramètres imbriqués, chacun sur sa propre ligne et indentés de deux espaces.

L'exemple suivant utilise une `services` définition pour configurer le point de terminaison à utiliser pour les demandes adressées uniquement au Amazon DynamoDB service. La "`local-dynamodb`" `services` section est référencée dans la `profile` section par son nom à l'aide de la `services` clé. La clé Service AWS d'identification est `dynamodb`. La sous-section des Amazon DynamoDB services commence sur la ligne `dynamodb =` . Toutes les lignes indentées qui suivent immédiatement sont incluses dans cette sous-section et s'appliquent à ce service.

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


Emplacement du partage **config** et des **credentials** fichiers

Les fichiers partagés sont AWS `config` des `credentials` fichiers en texte brut contenant des informations de configuration pour les AWS SDK et les outils. Les fichiers résident localement dans votre environnement et sont utilisés automatiquement par le code du SDK ou par AWS CLI les commandes que vous exécutez dans cet environnement. Par exemple, sur votre propre ordinateur ou lors du développement sur une instance Amazon Elastic Compute Cloud.

Lorsque le SDK ou l'outil s'exécute, il vérifie la présence de ces fichiers et charge tous les paramètres de configuration disponibles. Si les fichiers n'existent pas déjà, un fichier de base est automatiquement créé par le SDK ou l'outil.

Par défaut, les fichiers se trouvent dans un dossier nommé `.aws` qui est placé dans votre dossier home ou dans celui de l'utilisateur.

Système d'exploitation	Emplacement et nom par défaut des fichiers
Linux et macOS	<code>~/.aws/config</code> <code>~/.aws/credentials</code>
Windows	<code>%USERPROFILE%\ .aws\config</code> <code>%USERPROFILE%\ .aws\credentials</code>

Résolution du répertoire personnel

`~`n'est utilisé pour la résolution du répertoire personnel que lorsqu'il :

- Démarre le chemin
- Est immédiatement suivi par `/` ou par un séparateur spécifique à la plate-forme. Sous Windows, `~/` et `~\` les deux se résolvent dans le répertoire de base.

Lors de la détermination du répertoire de base, les variables suivantes sont vérifiées :

- (Toutes les plateformes) La variable d'`HOME`environnement
- (Plateformes Windows) La variable d'`USERPROFILE`environnement

- (Plateformes Windows) La concaténation de variables d'HOMEPAThenvironnement HOMEDRIVE et de variables d'environnement () \$HOMEDRIVE\$HOMEPATH
- (Facultatif par SDK ou outil) Fonction ou variable de résolution du chemin d'accueil spécifique au SDK ou à l'outil

Dans la mesure du possible, si le répertoire personnel d'un utilisateur est spécifié au début du chemin (par exemple,~username/), il est résolu dans le répertoire personnel du nom d'utilisateur demandé (par exemple,/home/username/.aws/config).

Modifier l'emplacement par défaut de ces fichiers

Vous pouvez utiliser l'une des méthodes suivantes pour modifier l'emplacement à partir duquel ces fichiers sont chargés par le SDK ou l'outil.

Utiliser des variables d'environnement

Les variables d'environnement suivantes peuvent être définies pour modifier l'emplacement ou le nom de ces fichiers de la valeur par défaut à une valeur personnalisée :

- configvariable d'environnement de fichier : **AWS_CONFIG_FILE**
- credentialvariable d'environnement de fichier : **AWS_SHARED_CREDENTIALS_FILE**

Linux/macOS

Vous pouvez spécifier un autre emplacement en exécutant les commandes [d'exportation](#) suivantes sous Linux ou 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

Vous pouvez spécifier un autre emplacement en exécutant les commandes [setx](#) suivantes sous 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
```

Pour plus d'informations sur la configuration de votre système à l'aide de variables d'environnement, consultez [Support des variables d'environnement](#).

Utiliser les propriétés du système JVM

Pour le SDK pour Kotlin exécuté sur la JVM et pour le SDK for Java 2.x, vous pouvez définir les propriétés du système JVM suivantes pour modifier l'emplacement ou le nom de ces fichiers de la valeur par défaut à une valeur personnalisée :

- `config` propriété du système JVM de fichiers : **`aws.configFile`**
- `credentials` variable d'environnement de fichier : **`aws.sharedCredentialsFile`**

Pour obtenir des instructions sur la façon de définir les propriétés du système JVM, consultez [the section called "Comment définir les propriétés du système JVM"](#). Le SDK pour Java 1.x ne prend pas en charge ces propriétés système.

Support des variables d'environnement

Les variables d'environnement représentent un autre moyen de spécifier les options de configuration et les informations d'identification, et peuvent être utiles pour les scripts ou la définition temporaire d'un profil nommé comme profil par défaut. Pour obtenir la liste des variables d'environnement prises en charge par la plupart des SDK, consultez [Liste des variables d'environnement](#).

Priorité d'options

- Si vous spécifiez un paramètre à l'aide de sa variable d'environnement, il remplace toute valeur chargée à partir d'un profil dans le partage AWS config et `credentials` les fichiers.
- Si vous spécifiez un paramètre à l'aide d'un paramètre sur la ligne de commande AWS CLI, il remplace toute valeur de la variable d'environnement correspondante ou d'un profil du fichier de configuration.

Comment définir les variables d'environnement

Les exemples suivants montrent comment vous pouvez configurer des variables d'environnement pour l'utilisateur par défaut.

Linux, macOS, or Unix

```
$ export AWS_ACCESS_KEY_ID=AKIAIOSFODNN7EXAMPLE
$ export AWS_SECRET_ACCESS_KEY=wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY
$ export
  AWS_SESSION_TOKEN=AQoEXAMPLEH4aoAH0gNCAPy...truncated...zrkuWJ0gQs8IZZaIv2BXIa2R40Lgk
$ export AWS_REGION=us-west-2
```

La définition de la variable d'environnement permet de modifier la valeur utilisée jusqu'à la fin de votre session shell, ou jusqu'à ce que vous définissiez la variable sur une autre valeur. Vous pouvez rendre les variables persistantes dans de futures sessions en les définissant dans votre script de démarrage de 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...zrkuWJ0gQs8IZZaIv2BXIa2R40Lgk
C:\> setx AWS_REGION us-west-2
```

L'utilisation [set](#) pour définir une variable d'environnement modifie la valeur utilisée jusqu'à la fin de la session d'invite de commande en cours ou jusqu'à ce que vous définissiez une valeur différente pour la variable. Le fait [setx](#) de définir une variable d'environnement modifie la valeur utilisée à la fois dans la session d'invite de commande en cours et dans toutes les sessions d'invite de commandes que vous créez après avoir exécuté la commande. Cela n'affecte pas les autres shells de commande qui sont déjà en cours d'exécution lorsque vous exécutez la commande.

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...zrkuWJ0gQs8IZZaIv2BXIa2R40Lgk"
PS C:\> $Env:AWS_REGION="us-west-2"
```

Si vous définissez une variable d'environnement à l'invite PowerShell, comme indiqué dans les exemples précédents, elle enregistre la valeur uniquement pendant la durée de la session en cours. Pour que le paramètre de variable d'environnement soit persistant dans toutes les sessions PowerShell et dans les sessions d'invite de commande, stockez-le à l'aide de l'application

système du Panneau de configuration. Vous pouvez également définir la variable pour toutes les PowerShell sessions futures en l'ajoutant à votre PowerShell profil. Consultez la [PowerShell documentation](#) pour plus d'informations sur le stockage des variables d'environnement ou leur persistance d'une session à l'autre.

Configuration de variables d'environnement sans serveur

Si vous utilisez une architecture sans serveur pour le développement, vous disposez d'autres options pour définir les variables d'environnement. En fonction de votre conteneur, vous pouvez utiliser différentes stratégies pour exécuter le code dans ces conteneurs afin de voir et d'accéder aux variables d'environnement, comme dans les environnements non cloud.

Par exemple, avec AWS Lambda, vous pouvez définir directement des variables d'environnement. Pour plus de détails, consultez la section [Utilisation de variables d'AWS Lambda environnement](#) dans le Guide du AWS Lambda développeur.

Dans Serverless Framework, vous pouvez souvent définir des variables d'environnement du SDK dans le `serverless.yml` fichier sous la clé du fournisseur sous le paramètre d'environnement. Pour plus d'informations sur le `serverless.yml` fichier, consultez la section [Paramètres généraux des fonctions](#) dans la documentation du Serverless Framework.

Quel que soit le mécanisme que vous utilisez pour définir les variables d'environnement du conteneur, certaines sont réservées par le conteneur, comme celles décrites pour Lambda dans la section Variables [d'environnement d'exécution définies](#). Consultez toujours la documentation officielle du conteneur que vous utilisez pour déterminer comment les variables d'environnement sont traitées et s'il existe des restrictions.

Support des propriétés du système JVM

[Les propriétés du système JVM](#) fournissent un autre moyen de spécifier les options de configuration et les informations d'identification pour les SDK qui s'exécutent sur la JVM, tels que le et le AWS SDK for Java . Kit AWS SDK pour Kotlin Pour obtenir la liste des propriétés du système JVM prises en charge par les SDK, consultez la section Référence des [paramètres](#).

Priorité d'options

- Si vous spécifiez un paramètre à l'aide de sa propriété système JVM, il remplace toute valeur trouvée dans les variables d'environnement ou chargée à partir d'un profil dans l'AWS `config` et `credentials` les fichiers partagés.

- Si vous spécifiez un paramètre à l'aide de sa variable d'environnement, il remplace toute valeur chargée à partir d'un profil dans l'AWS config et `credentials` les fichiers partagés.

Comment définir les propriétés du système JVM

Vous pouvez définir les propriétés du système JVM de plusieurs manières.

Sur la ligne de commande

Définissez les propriétés du système JVM sur la ligne de commande lorsque vous appelez la `java` commande à l'aide du commutateur. `-D` La commande suivante configure le Région AWS globalement pour tous les clients du service, sauf si vous remplacez explicitement la valeur dans le code.

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

Si vous devez définir plusieurs propriétés du système JVM, spécifiez le `-D` commutateur plusieurs fois.

Avec une variable d'environnement

Si vous ne pouvez pas accéder à la ligne de commande pour appeler la machine virtuelle Java afin d'exécuter votre application, vous pouvez utiliser la variable d'`JAVA_TOOL_OPTIONS` environnement pour configurer les options de ligne de commande. Cette approche est utile dans des situations telles que l'exécution d'une AWS Lambda fonction sur le runtime Java ou l'exécution de code dans une machine virtuelle Java intégrée.

L'exemple suivant configure le Région AWS globalement pour tous les clients du service, sauf si vous remplacez explicitement la valeur dans le code.

Linux, macOS, or Unix

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

La définition de la variable d'environnement permet de modifier la valeur utilisée jusqu'à la fin de votre session shell, ou jusqu'à ce que vous définissiez la variable sur une autre valeur. Vous pouvez rendre les variables persistantes dans de futures sessions en les définissant dans votre script de démarrage de shell.

Windows Command Prompt

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

L'utilisation [set](#) pour définir une variable d'environnement modifie la valeur utilisée jusqu'à la fin de la session d'invite de commande en cours ou jusqu'à ce que vous définissiez une valeur différente pour la variable. Le fait [setx](#) de définir une variable d'environnement modifie la valeur utilisée à la fois dans la session d'invite de commande en cours et dans toutes les sessions d'invite de commandes que vous créez après avoir exécuté la commande. Cela n'affecte pas les autres shells de commande qui sont déjà en cours d'exécution lorsque vous exécutez la commande.

Au moment de l'exécution

Vous pouvez également définir les propriétés du système JVM lors de l'exécution dans le code en utilisant la `System.setProperty` méthode illustrée dans l'exemple suivant.

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

Important

Définissez les propriétés du système JVM avant d'initialiser les clients du service SDK, sinon les clients de service peuvent utiliser d'autres valeurs.

Authentification et accès

Vous devez définir la manière dont votre code s'authentifie AWS lorsque vous développez avec Services AWS. Vous pouvez configurer l'accès programmatique aux AWS ressources de différentes manières, en fonction de l'environnement et de l'AWS accès dont vous disposez.

Options d'authentification pour le code exécuté localement (pas dans AWS)

- [Authentification IAM Identity Center](#)— En tant que bonne pratique en matière de sécurité, nous vous recommandons AWS Organizations d'utiliser IAM Identity Center pour gérer l'accès de tous vos Comptes AWS utilisateurs. Vous pouvez créer des utilisateurs dans AWS IAM Identity Center, utiliser Microsoft Active Directory, utiliser un fournisseur d'identité (IdP) SAML 2.0 ou fédérer individuellement votre IdP avec. Comptes AWS Pour vérifier si votre région prend en charge le centre d'identité IAM, consultez la section sur les [AWS IAM Identity Center points de terminaison et les quotas](#) dans le. Référence générale d'Amazon Web Services
- [Rôles Anywhere IAM](#)— Vous pouvez utiliser IAM Roles Anywhere pour obtenir des informations d'identification de sécurité temporaires dans IAM pour les charges de travail telles que les serveurs, les conteneurs et les applications qui s'exécutent en dehors de. AWS Pour utiliser IAM Roles Anywhere, vos charges de travail doivent utiliser des certificats X.509.
- [Assumer un rôle](#)— Vous pouvez assumer un rôle IAM pour accéder temporairement à AWS des ressources auxquelles vous n'auriez peut-être pas accès autrement.
- [AWS clés d'accès](#)— D'autres options peuvent être moins pratiques ou augmenter les risques de sécurité pour vos AWS ressources.

Options d'authentification pour le code exécuté dans un AWS environnement

- [Utilisation de rôles IAM pour les instances Amazon EC2](#)— Utilisez les rôles IAM pour exécuter votre application en toute sécurité sur une instance Amazon EC2.
- Vous pouvez interagir par programmation avec AWS IAM Identity Center de la manière suivante :
 - Permet [AWS CloudShell](#) d'exécuter AWS CLI des commandes depuis la console.
 - [AWS Cloud9](#) À utiliser pour commencer à programmer en AWS utilisant un environnement de développement intégré (IDE) avec AWS des ressources.
 - Pour essayer un espace de collaboration basé sur le cloud pour les équipes de développement de logiciels, pensez à utiliser [Amazon CodeCatalyst](#).

Authentification via un fournisseur d'identité basé sur le Web - Applications Web mobiles ou basées sur le client

Si vous créez des applications mobiles ou des applications Web basées sur des clients qui nécessitent un accès à AWS, créez votre application de manière à ce qu'elle demande des informations d'identification de AWS sécurité temporaires de manière dynamique à l'aide de la fédération d'identité Web.

Lors de l'utilisation de la fédération d'identité web, il n'est pas nécessaire de créer de code de connexion personnalisé ni de gérer vos propres identités utilisateur. Les utilisateurs de l'application peuvent plutôt se connecter à l'aide d'un fournisseur d'identité externe (IdP) connu, tel que Login with Amazon, Facebook, Google ou tout autre IdP compatible avec OpenID Connect (OIDC). Ils peuvent recevoir un jeton d'authentification, puis échanger ce jeton contre des informations d'identification de sécurité temporaires dans AWS qui sont mappées à un rôle IAM disposant d'autorisations permettant d'utiliser les ressources de votre Compte AWS.

Pour savoir comment le configurer pour votre SDK ou votre outil, consultez [Fédérez avec l'identité Web ou OpenID Connect](#).

Pour les applications mobiles, pensez à utiliser Amazon Cognito. Amazon Cognito agit en tant que courtier d'identité et effectue une grande partie du travail de fédération à votre place. Pour plus d'informations, consultez la section [Utilisation d'Amazon Cognito pour les applications mobiles](#) dans le guide de l'utilisateur IAM.

Plus d'informations sur la gestion des accès

Le guide de l'utilisateur IAM contient les informations suivantes sur le contrôle sécurisé de l'accès aux AWS ressources :

- [Identités IAM \(utilisateurs, groupes d'utilisateurs et rôles\)](#) : comprenez les bases des identités dans AWS.
- [Bonnes pratiques de sécurité en matière d'IAM](#) : recommandations de sécurité à suivre lors du développement d'AWS applications selon le modèle de [responsabilité partagée](#).

Référence générale d'Amazon Web Services Il contient des éléments de base sur les points suivants :

- [Comprendre et obtenir vos AWS informations d'identification](#) : accédez aux principales options et pratiques de gestion pour l'accès par console et par programmation.

ID de constructeur AWS

Vos ID de constructeur AWS compléments à ceux Comptes AWS que vous possédez déjà ou que vous souhaitez créer. Alors qu'un Compte AWS agit comme un conteneur pour les AWS ressources que vous créez et fournit une limite de sécurité pour ces ressources, vous vous ID de constructeur AWS représente en tant qu'individu. Vous pouvez vous connecter ID de constructeur AWS pour accéder à des outils et services de développement tels qu'Amazon CodeWhisperer et Amazon CodeCatalyst.

- [ID de constructeur AWS](#) [Connectez-vous avec](#) le guide de l'Connexion à AWS utilisateur : découvrez comment créer et utiliser un Builder ID ID de constructeur AWS et découvrez ce que fournit le Builder ID.
- [Authentification avec CodeWhisperer et AWS Toolkit - Builder ID](#) dans le guide de CodeWhisperer l'utilisateur — Découvrez comment CodeWhisperer utilise un ID de constructeur AWS.
- [CodeCatalyst concepts - ID de constructeur AWS](#) dans le guide de CodeCatalyst l'utilisateur Amazon - Découvrez comment CodeCatalyst utilise un ID de constructeur AWS.

Authentification IAM Identity Center

AWS IAM Identity Center est la méthode recommandée pour fournir des AWS informations d'identification lors du développement sur un service non AWS informatique. Par exemple, cela pourrait être quelque chose comme votre environnement de développement local. Si vous développez sur une AWS ressource, telle qu'Amazon Elastic Compute Cloud (Amazon EC2) AWS Cloud9, nous vous recommandons d'obtenir des informations d'identification auprès de ce service.

Dans ce didacticiel, vous établissez l'accès à IAM Identity Center et vous le configurez pour votre SDK ou outil à l'aide du portail AWS d'accès et du. AWS CLI

- Le portail AWS d'accès est l'emplacement Web où vous vous connectez manuellement à l'IAM Identity Center. Le format de l'URL est `d-xxxxxxxxxx.awsapps.com/start` ou `your_subdomain.awsapps.com/start`. Lorsque vous êtes connecté au portail AWS d'accès, vous pouvez consulter Comptes AWS les rôles qui ont été configurés pour cet utilisateur. Cette procédure utilise le portail AWS d'accès pour obtenir les valeurs de configuration dont vous avez besoin pour le processus d'authentification du SDK ou de l'outil.
- AWS CLI II est utilisé pour configurer votre SDK ou votre outil afin d'utiliser l'authentification IAM Identity Center pour les appels d'API effectués par votre code. Ce processus unique met à jour

vosre AWS config fichier partagé, qui est ensuite utilisé par votre SDK ou votre outil lorsque vous exécutez votre code.

Configuration de l'accès programmatique à l'aide d'IAM Identity Center

Étape 1 : établir l'accès et sélectionner l'ensemble d'autorisations approprié

Si vous n'avez pas encore activé IAM Identity Center, consultez la section [Activation d'IAM Identity Center](#) dans le guide de l'AWS IAM Identity Center utilisateur.

Choisissez l'une des méthodes suivantes pour accéder à vos AWS informations d'identification.

Je n'ai pas établi d'accès via IAM Identity Center

1. Ajoutez un utilisateur et ajoutez des autorisations administratives en suivant la procédure de [configuration de l'accès utilisateur avec le répertoire IAM Identity Center par défaut](#) du guide de l'AWS IAM Identity Center utilisateur.
2. L'ensemble `AdministratorAccess` d'autorisations ne doit pas être utilisé pour le développement normal. Nous vous recommandons plutôt d'utiliser l'ensemble d'autorisations `PowerUserAccess` prédéfini, sauf si votre employeur a créé un ensemble d'autorisations personnalisé à cette fin.

Suivez à nouveau la même procédure de [configuration de l'accès utilisateur avec la procédure d'annuaire par défaut d'IAM Identity Center](#), mais cette fois :

- Au lieu de créer le *Admin team* groupe, créez-en un *Dev team* et remplacez-le par la suite dans les instructions.
- Vous pouvez utiliser l'utilisateur existant, mais celui-ci doit être ajouté au nouveau *Dev team* groupe.
- Au lieu de créer l'ensemble d'autorisations `AdministratorAccess`, créez-en un `PowerUserAccess` et remplacez-le par la suite dans les instructions.

Lorsque vous aurez terminé, vous devriez avoir les éléments suivants :

- Un `Dev team` groupe.
- Une `PowerUserAccess` autorisation attachée définie pour le `Dev team` groupe.
- Votre utilisateur a été ajouté au `Dev team` groupe.

3. Quittez le portail et reconnectez-vous pour voir vos options Comptes AWS et celles pour `Administrator` ou `PowerUserAccess`. Sélectionnez `PowerUserAccess` lorsque vous travaillez avec votre outil/SDK.

J'y ai déjà accès AWS via un fournisseur d'identité fédéré géré par mon employeur (tel que Microsoft Entra ou Okta)

Connectez-vous AWS via le portail de votre fournisseur d'identité. Si votre administrateur cloud vous a accordé des autorisations `PowerUserAccess` (de développeur), vous voyez Comptes AWS celles auxquelles vous avez accès et votre ensemble d'autorisations. À côté du nom de votre ensemble d'autorisations, vous trouverez des options permettant d'accéder aux comptes manuellement ou par programmation à l'aide de cet ensemble d'autorisations.

Les implémentations personnalisées peuvent entraîner des expériences différentes, telles que des noms d'ensembles d'autorisations différents. Si vous ne savez pas quel ensemble d'autorisations utiliser, contactez votre équipe informatique pour obtenir de l'aide.

J'y ai déjà accès AWS via le portail AWS d'accès géré par mon employeur

Connectez-vous AWS via le portail AWS d'accès. Si votre administrateur cloud vous a accordé des autorisations `PowerUserAccess` (de développeur), vous voyez Comptes AWS celles auxquelles vous avez accès et votre ensemble d'autorisations. À côté du nom de votre ensemble d'autorisations, vous trouverez des options permettant d'accéder aux comptes manuellement ou par programmation à l'aide de cet ensemble d'autorisations.

J'y ai déjà accès AWS via un fournisseur d'identité personnalisé fédéré géré par mon employeur

Contactez votre équipe informatique pour obtenir de l'aide.

Étape 2 : Configuration des SDK et des outils pour utiliser IAM Identity Center

1. Sur votre machine de développement, installez la dernière version AWS CLI.
 - a. Consultez la section [Installation ou mise à jour de la dernière version du AWS CLI](#) dans le guide de AWS Command Line Interface l'utilisateur.
 - b. (Facultatif) Pour vérifier que le AWS CLI fonctionne, ouvrez une invite de commande et exécutez la `aws --version` commande.
2. Connectez-vous au portail d' AWS accès. Votre employeur peut fournir cette URL ou vous pouvez l'obtenir par e-mail après l'étape 1 : Établir l'accès. Si ce n'est pas le cas, trouvez

l'URL de votre portail d'AWS accès sur le tableau de bord de <https://console.aws.amazon.com/singlesignon/>.

- a. Dans le portail AWS d'accès, dans l'onglet Comptes, sélectionnez le compte individuel à gérer. Les rôles de votre utilisateur sont affichés. Choisissez les clés d'accès pour obtenir les informations d'identification pour la ligne de commande ou l'accès programmatique pour l'ensemble d'autorisations approprié. Utilisez l'ensemble `PowerUserAccess` d'autorisations prédéfini, ou le jeu d'autorisations que vous ou votre employeur avez créé pour appliquer les autorisations du moindre privilège au développement.
 - b. Dans la boîte de dialogue Obtenir les informations d'identification, choisissez macOS et Linux ou Windows, selon votre système d'exploitation.
 - c. Choisissez la méthode d'identification IAM Identity Center pour obtenir les SSO `Region` valeurs SSO `Start URL` et dont vous avez besoin pour l'étape suivante.
3. Dans l'invite de AWS CLI commande, exécutez la `aws configure sso` commande. Lorsque vous y êtes invité, entrez les valeurs de configuration que vous avez collectées à l'étape précédente. Pour plus de détails sur cette AWS CLI commande, voir [Configurer votre profil à l'aide de l'aws configure ssoassistant](#).
- Pour le nom du profil CLI, nous vous recommandons de saisir la *valeur par défaut* lorsque vous démarrez. Pour plus d'informations sur la façon de définir des profils (nommés) autres que ceux par défaut et leur variable d'environnement associée, consultez [Profils](#).
4. (Facultatif) Dans l'invite de AWS CLI commande, confirmez l'identité de la session active en exécutant la `aws sts get-caller-identity` commande. La réponse doit indiquer l'ensemble d'autorisations IAM Identity Center que vous avez configuré.
5. Si vous utilisez un AWS SDK, créez une application pour celui-ci dans votre environnement de développement.
- a. Pour certains SDK, des packages supplémentaires tels que SSO et SSO0IDC doivent être ajoutés à votre application avant de pouvoir utiliser l'authentification IAM Identity Center. Pour plus de détails, consultez votre SDK spécifique.
 - b. Si vous avez déjà configuré l'accès à AWS, passez en revue votre `AWS credentials` fichier partagé pour en détecter d'éventuels [AWS clés d'accès](#). Vous devez supprimer toutes les informations d'identification statiques avant que le SDK ou l'outil n'utilise les informations d'identification du centre d'identité IAM en raison de la [Chaîne de fournisseurs d'identifiants](#) priorité.

Pour en savoir plus sur la façon dont les SDK et les outils utilisent et actualisent les informations d'identification à l'aide de cette configuration, voir [Comprendre l'authentification IAM Identity Center](#).

En fonction de la durée de session que vous avez configurée, votre accès finira par expirer et le SDK ou l'outil rencontrera une erreur d'authentification. Pour actualiser à nouveau la session du portail d'accès en cas de besoin, utilisez la commande AWS CLI pour exécuter la `aws sso login` commande.

Vous pouvez prolonger à la fois la durée de session du portail d'accès IAM Identity Center et celle de la session d'ensemble d'autorisations. Cela allonge le délai pendant lequel vous pouvez exécuter le code avant de devoir vous reconnecter manuellement à l'aide du AWS CLI. Pour plus d'informations, consultez les rubriques suivantes dans le AWS IAM Identity Center Guide de l'utilisateur :

- Durée de session IAM Identity Center : [configurez la durée des sessions du portail d' AWS accès de vos utilisateurs](#)
- Autoriser définir la durée de la session — [Définir la durée de la session](#)

Pour plus de détails sur tous les paramètres du fournisseur IAM Identity Center pour les SDK et les outils, consultez ce [Fournisseur d'identifiants IAM Identity Center](#) guide.

Comprendre l'authentification IAM Identity Center

Termes pertinents du centre d'identité IAM

Les termes suivants vous aident à comprendre le processus et la configuration sous-jacents AWS IAM Identity Center. La documentation des API du AWS SDK utilise des noms différents de ceux d'IAM Identity Center pour certains de ces concepts d'authentification. Il est utile de connaître les deux noms.

Le tableau suivant montre comment les noms alternatifs sont liés les uns aux autres.

Nom du centre d'identité IAM	Nom de l'API du SDK	Description
Centre d'identité	sso	Bien que AWS Single Sign-On soit renommé, les espaces de noms de l'sso API conserveront leur nom d'origine à des fins de rétrocompatibilité.

Nom du centre d'identité IAM	Nom de l'API du SDK	Description
		Pour plus d'informations, voir Renommer le centre d'identité IAM dans le guide de l'AWS IAM Identity Center utilisateur.
Console IAM Identity Center Console d'administration		La console que vous utilisez pour configurer l'authentification unique.
AWSURL du portail d'accès		Une URL propre à votre compte IAM Identity Center, par exemple <code>https://xxx.awsapps.com/start</code> . Vous vous connectez à ce portail à l'aide de vos identifiants de connexion IAM Identity Center.
Session sur le portail d'accès à l'IAM Identity Center	Session d'authentification	Fournit un jeton d'accès au porteur à l'appelant.
Session d'ensemble d'autorisations		La session IAM que le SDK utilise en interne pour effectuer les Service AWS appels. Dans les discussions informelles, il est possible que cette session soit incorrectement appelée « session de rôle ».

Nom du centre d'identité IAM	Nom de l'API du SDK	Description
Informations d'identification du jeu d'autorisations	AWSInformations d'identification informations d'identification sigv4	Les informations d'identification que le SDK utilise réellement pour la plupart des Service AWS appels (en particulier, tous les Service AWS appels sigv4). Dans les discussions informelles, il est possible que cela soit appelé à tort « informations d'identification du rôle ».
Fournisseur d'identifiants IAM Identity Center	Fournisseur d'informations d'identification SSO	Comment obtenez-vous les informations d'identification, telles que la classe ou le module fournissant les fonctionnalités.

Comprendre la résolution des informations d'identification du SDK pour Services AWS

L'API IAM Identity Center échange les informations d'identification du jeton porteur contre des informations d'identification sigv4. La plupart Services AWS sont des API sigv4, à quelques exceptions près, comme Amazon CodeWhisperer et Amazon CodeCatalyst. Ce qui suit décrit le processus de résolution des informations d'identification permettant de prendre en charge la plupart des Service AWS appels visant à obtenir le code de votre application. AWS IAM Identity Center

Démarrage d'une session de portail d'accès AWS

- Commencez le processus en vous connectant à la session avec vos informations d'identification.
 - Utilisez la `aws sso login` commande dans le AWS Command Line Interface (AWS CLI). Cela démarre une nouvelle session IAM Identity Center si vous n'avez pas encore de session active.
- Lorsque vous démarrez une nouvelle session, vous recevez un jeton d'actualisation et un jeton d'accès de la part d'IAM Identity Center. AWS CLI met également à jour un fichier JSON de cache SSO avec un nouveau jeton d'accès et un nouveau jeton d'actualisation et le rend disponible pour utilisation par les SDK.

- Si vous avez déjà une session active, la AWS CLI commande réutilise la session existante et expirera chaque fois que la session existante expirera. Pour savoir comment définir la durée d'une session IAM Identity Center, voir [Configurer la durée des sessions du portail d'AWS Accès de vos utilisateurs](#) dans le Guide de l'AWS IAM Identity Center utilisateur.
- La durée maximale des sessions a été étendue à 90 jours afin de réduire le besoin de connexions fréquentes.

Comment le SDK obtient les informations d'identification pour les appels Service AWS

Les SDK fournissent un accès Services AWS lorsque vous instanciez un objet client par service. Lorsque le profil sélectionné du AWS config fichier partagé est configuré pour la résolution des informations d'identification IAM Identity Center, IAM Identity Center est utilisé pour résoudre les informations d'identification de votre application.

- Le [processus de résolution des informations d'identification](#) est terminé pendant l'exécution lorsqu'un client est créé.

Pour récupérer les informations d'identification des API sigv4 à l'aide de l'authentification unique IAM Identity Center, le SDK utilise le jeton d'accès IAM Identity Center pour obtenir une session IAM. Cette session IAM est appelée session d'ensemble d'autorisations et permet d'AWS accéder au SDK en assumant un rôle IAM.

- La durée de session de l'ensemble d'autorisations est définie indépendamment de celle de la session IAM Identity Center.
 - Pour savoir comment définir la durée de session définie par les autorisations, voir [Définir la durée de session](#) dans le guide de AWS IAM Identity Center l'utilisateur.
- Sachez que les informations d'identification de l'ensemble d'autorisations sont également appelées informations AWS d'identification et informations d'identification sigv4 dans la plupart des documentations d'API du AWS SDK.

Les informations d'identification de l'ensemble d'autorisations sont renvoyées par un appel [getRoleCredentials](#) de l'API IAM Identity Center au SDK. L'objet client du SDK utilise ce rôle IAM supposé pour effectuer des appels Service AWS, par exemple pour demander à Amazon S3 de répertorier les buckets de votre compte. L'objet client peut continuer à fonctionner en utilisant ces informations d'identification du jeu d'autorisations jusqu'à l'expiration de la session du jeu d'autorisations.

Expiration et actualisation de la session

Lorsque vous utilisez le [Configuration du fournisseur de jetons SSO](#), le jeton d'accès horaire obtenu auprès d'IAM Identity Center est automatiquement actualisé à l'aide du jeton d'actualisation.

- Si le jeton d'accès a expiré lorsque le SDK essaie de l'utiliser, le SDK utilise le jeton d'actualisation pour essayer d'obtenir un nouveau jeton d'accès. L'IAM Identity Center compare le jeton d'actualisation à la durée de la session de votre portail d'accès IAM Identity Center. Si le jeton d'actualisation n'est pas expiré, le centre d'identité IAM répond avec un autre jeton d'accès.
- Ce jeton d'accès peut être utilisé soit pour actualiser la session d'ensemble d'autorisations des clients existants, soit pour résoudre les informations d'identification des nouveaux clients.

Toutefois, si la session du portail d'accès IAM Identity Center est expirée, aucun nouveau jeton d'accès n'est accordé. Par conséquent, la durée définie d'autorisations ne peut pas être renouvelée. Il expirera (et l'accès sera perdu) chaque fois que la durée de session définie par le cache expirera pour les clients existants.

Tout code qui crée un nouveau client échouera à l'authentification dès l'expiration de la session IAM Identity Center. Cela est dû au fait que les informations d'identification du jeu d'autorisations ne sont pas mises en cache. Votre code ne sera pas en mesure de créer un nouveau client et de terminer le processus de résolution des informations d'identification tant que vous ne disposerez pas d'un jeton d'accès valide.

Pour récapituler, lorsque le SDK a besoin de nouvelles informations d'identification d'un ensemble d'autorisations, le SDK vérifie d'abord si des informations d'identification existantes sont valides et les utilise. Cela s'applique qu'il s'agisse d'un nouveau client ou d'un client existant dont les informations d'identification ont expiré. Si les informations d'identification ne sont pas trouvées ou si elles ne sont pas valides, le SDK appelle l'API IAM Identity Center pour obtenir de nouvelles informations d'identification. Pour appeler l'API, elle a besoin du jeton d'accès. Si le jeton d'accès est expiré, le SDK utilise le jeton d'actualisation pour essayer d'obtenir un nouveau jeton d'accès auprès du service IAM Identity Center. Ce jeton est accordé si votre session du portail d'accès à l'IAM Identity Center n'a pas expiré.

Rôles Anywhere IAM

Vous pouvez utiliser IAM Roles Anywhere pour obtenir des informations d'identification de sécurité temporaires dans IAM pour les charges de travail telles que les serveurs, les conteneurs et les applications qui s'exécutent en dehors de. AWS Pour utiliser IAM Roles Anywhere, vos charges de

travail doivent utiliser des certificats X.509. Votre administrateur cloud doit fournir le certificat et la clé privée nécessaires pour configurer IAM Roles Anywhere en tant que fournisseur d'informations d'identification.

Étape 1 : configurer les rôles IAM n'importe où

IAM Roles Anywhere permet d'obtenir des informations d'identification temporaires pour une charge de travail ou un processus qui s'exécute en dehors de AWS. Une ancre de confiance est établie avec l'autorité de certification afin d'obtenir des informations d'identification temporaires pour le rôle IAM associé. Le rôle définit les autorisations dont bénéficiera votre charge de travail lorsque votre code s'authentifie auprès d'IAM Roles Anywhere.

Pour connaître les étapes de configuration de l'ancre de confiance, du rôle IAM et du profil IAM Roles Anywhere, consultez la section [Création d'une ancre de confiance et d'un profil dans AWS Identity and Access Management Roles Anywhere](#) du guide de l'utilisateur d'IAM Roles Anywhere.

Note

Un profil figurant dans le guide de l'utilisateur d'IAM Roles Anywhere fait référence à un concept unique au sein du service IAM Roles Anywhere. Cela n'est pas lié aux profils contenus dans le AWS config fichier partagé.

Étape 2 : Utiliser les rôles IAM n'importe où

Pour obtenir des informations d'identification de sécurité temporaires auprès d'IAM Roles Anywhere, utilisez l'outil d'aide aux informations d'identification fourni par IAM Roles Anywhere. L'outil d'identification met en œuvre le processus de signature pour IAM Roles Anywhere.

Pour obtenir des instructions sur le téléchargement de l'outil d'aide aux informations d'identification, consultez la section [Obtention d'informations d'identification de sécurité temporaires auprès de AWS Identity and Access Management Roles Anywhere](#) dans le guide de l'utilisateur d'IAM Roles Anywhere.

Pour utiliser les informations d'identification de sécurité temporaires d'IAM Roles Anywhere avec AWS les SDK et le AWS CLI, vous pouvez configurer les `credential_process` paramètres dans le fichier partagé `AWSconfig`. Les SDK et le AWS CLI support sont un fournisseur d'informations d'identification de processus utilisé `credential_process` pour l'authentification. Ce qui suit montre la structure générale à définir `credential_process`.

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

La `credential-process` commande de l'outil d'assistance renvoie des informations d'identification temporaires dans un format JSON standard compatible avec le `credential_process` paramètre. Notez que le nom de la commande contient un trait d'union mais que le nom du paramètre contient un trait de soulignement. La commande nécessite les paramètres suivants :

- `private-key`— Le chemin d'accès à la clé privée qui a signé la demande.
- `certificate`— Le chemin d'accès au certificat.
- `role-arn`— L'ARN du rôle pour lequel vous souhaitez obtenir des informations d'identification temporaires.
- `profile-arn`— L'ARN du profil qui fournit un mappage pour le rôle spécifié.
- `trust-anchor-arn`— L'ARN de l'ancre de confiance utilisée pour l'authentification.

Votre administrateur cloud doit fournir le certificat et la clé privée. Les trois valeurs ARN peuvent être copiées depuis le AWS Management Console. L'exemple suivant montre un config fichier partagé qui configure la récupération d'informations d'identification temporaires à partir de l'outil d'assistance.

```
[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
```

Pour les paramètres facultatifs et des informations supplémentaires sur les outils d'assistance, voir [IAM Roles Anywhere Credential Helper](#) on. GitHub

Pour plus de détails sur le paramètre de configuration du SDK lui-même et sur le fournisseur d'informations d'identification du processus, consultez ce [Fournisseur d'identifiants de processus](#) guide.

Assumer un rôle

Assumer un rôle implique l'utilisation d'un ensemble d'informations d'identification de sécurité temporaires pour accéder à AWS des ressources auxquelles vous n'auriez peut-être pas accès

autrement. Ces informations d'identification temporaires incluent un ID de clé d'accès, une clé d'accès secrète et un jeton de sécurité. Pour en savoir plus sur les demandes d'API AWS Security Token Service (AWS STS), consultez la section [Actions](#) de la référence AWS Security Token Service d'API.

Pour configurer votre SDK ou votre outil afin qu'il assume un rôle, vous devez d'abord créer ou identifier un rôle spécifique à assumer. Les rôles IAM sont identifiés de manière unique par un rôle Amazon Resource Name ([ARN](#)). Les rôles établissent des relations de confiance avec une autre entité. L'entité de confiance qui utilise le rôle peut être un Service AWS fournisseur d'identité WebCompte AWS, une fédération OIDC ou SAML. Pour en savoir plus sur les rôles IAM, consultez la section [Utilisation des rôles IAM](#) dans le Guide de l'utilisateur IAM.

Une fois le rôle IAM identifié, si ce rôle vous fait confiance, vous pouvez configurer votre SDK ou votre outil pour utiliser les autorisations accordées par le rôle. Pour ce faire, soit, [Assumez un rôle IAM](#) soit [Fédérez avec l'identité Web ou OpenID Connect](#).

Assumez un rôle IAM

Lorsque vous assumez un rôle, AWS STS renvoie un ensemble d'informations d'identification de sécurité temporaires. Ces informations d'identification proviennent d'un autre profil ou de l'instance ou du conteneur dans lequel votre code est exécuté. D'autres exemples d'attribution d'un rôle incluent la gestion Comptes AWS de plusieurs comptes depuis Amazon EC2, l'utilisation de AWS CodeCommit across Comptes AWS ou l'accès à un autre compte depuis Amazon EC2. AWS CodeBuild

Étape 1 : configurer un rôle IAM

Pour configurer votre SDK ou votre outil afin qu'il assume un rôle, vous devez d'abord créer ou identifier un rôle spécifique à assumer. Les rôles IAM sont identifiés de manière unique à l'aide d'un [ARN](#) de rôle. Les rôles établissent des relations de confiance avec une autre entité, généralement au sein de votre compte ou pour un accès entre comptes. Pour configurer cela, consultez la section [Création de rôles IAM](#) dans le guide de l'utilisateur IAM.

Étape 2 : Configuration du SDK ou de l'outil

Configurez le SDK ou l'outil pour obtenir les informations d'identification depuis `credential_source` ou `source_profile`.

`credential_source` À utiliser pour obtenir des informations d'identification à partir d'un conteneur Amazon ECS, d'une instance Amazon EC2 ou de variables d'environnement.

`source_profile` À utiliser pour obtenir des informations d'identification à partir d'un autre profil. `source_profile` prend également en charge le chaînage des rôles, qui consiste en des hiérarchies de profils dans lesquelles un rôle assumé est ensuite utilisé pour assumer un autre rôle.

Lorsque vous le spécifiez dans un profil, le SDK ou l'outil lance automatiquement l'appel d'AWS STS [AssumeRole](#) API correspondant pour vous. Pour récupérer et utiliser des informations d'identification temporaires en assumant un rôle, spécifiez les valeurs de configuration suivantes dans le AWS config fichier partagé. Pour plus de détails sur chacun de ces paramètres, consultez la [Assumer les paramètres du fournisseur d'informations d'identification du rôle](#) section.

- `role_arn`- À partir du rôle IAM que vous avez créé à l'étape 1
- Configurez `source_profile` soit `credential_source`
- (Facultatif) `duration_seconds`
- (Facultatif) `external_id`
- (Facultatif) `mfa_serial`
- (Facultatif) `role_session_name`

Les exemples suivants montrent la configuration des deux options d'attribution de rôle dans un config fichier partagé :

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

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

Pour plus de détails sur tous les paramètres du fournisseur d'informations d'identification d'assumer un rôle, consultez [Assumer le rôle de fournisseur d'informations d'identification](#) ce guide.

Fédérez avec l'identité Web ou OpenID Connect

Lorsque vous créez des applications mobiles ou des applications Web basées sur le client qui nécessitent un accès à AWS, AWS STS renvoie un ensemble d'informations d'identification de sécurité temporaires pour les utilisateurs fédérés authentifiés par le biais d'un fournisseur d'identité public (IdP). Login with Amazon, Facebook, Google ou tout fournisseur d'identité compatible avec OpenID Connect (OIDC) sont des exemples de fournisseurs d'identité publics. Avec cette méthode, vos utilisateurs n'ont pas besoin de leur propre identité AWS ou de leur identité IAM.

Si vous utilisez Amazon Elastic Kubernetes Service, cette fonctionnalité permet de spécifier différents rôles IAM pour chacun de vos conteneurs. Kubernetes permet de distribuer des jetons OIDC à vos conteneurs qui sont utilisés par ce fournisseur d'informations d'identification pour obtenir des informations d'identification temporaires. Pour plus d'informations sur cette configuration Amazon EKS, consultez la section [Rôles IAM pour les comptes de service](#) dans le guide de l'utilisateur Amazon EKS. Toutefois, pour une option plus simple, nous vous recommandons d'utiliser [Amazon EKS Pod Identities](#) à la place si votre [SDK le prend en charge](#).

Étape 1 : configurer un fournisseur d'identité et un rôle IAM

Pour configurer la fédération avec un IdP externe, utilisez un fournisseur d'identité IAM pour fournir des AWS informations sur l'IdP externe et sa configuration. Cela établit la confiance entre votre Compte AWS IdP et l'IdP externe. Avant de configurer le SDK pour utiliser le jeton d'identité Web pour l'authentification, vous devez d'abord configurer le fournisseur d'identité (IdP) et le rôle IAM utilisé pour y accéder. Pour les configurer, consultez la section [Création d'un rôle pour l'identité Web ou OpenID Connect Federation \(console\)](#) dans le guide de l'utilisateur IAM.

Étape 2 : Configuration du SDK ou de l'outil

Configurez le SDK ou l'outil pour utiliser un jeton d'identité Web à des AWS STS fins d'authentification.

Lorsque vous le spécifiez dans un profil, le SDK ou l'outil lance automatiquement l'appel d'AWS STS [AssumeRoleWithWebIdentity](#) API correspondant pour vous. Pour récupérer et utiliser des informations d'identification temporaires à l'aide de la fédération d'identité Web, spécifiez les valeurs de configuration suivantes dans le AWS config fichier partagé. Pour plus de détails sur chacun de ces paramètres, consultez la [Assumer les paramètres du fournisseur d'informations d'identification du rôle](#) section.

- `role_arn`- À partir du rôle IAM que vous avez créé à l'étape 1
- `web_identity_token_file`- Depuis l'IdP externe
- (Facultatif) `duration_seconds`
- (Facultatif) `role_session_name`

Voici un exemple de configuration de config fichier partagé pour assumer un rôle avec une identité Web :

```
[profile web-identity]
```

```
role_arn=arn:aws:iam::123456789012:role/my-role-name  
web_identity_token_file=/path/to/a/token
```

Note

Pour les applications mobiles, pensez à utiliser Amazon Cognito. Amazon Cognito agit en tant que courtier d'identité et effectue une grande partie du travail de fédération à votre place. Cependant, le fournisseur d'identité Amazon Cognito n'est pas inclus dans les bibliothèques principales des SDK et des outils comme les autres fournisseurs d'identité. Pour accéder à l'API Amazon Cognito, incluez le client du service Amazon Cognito dans la version ou les bibliothèques de votre SDK ou outil. Pour une utilisation avec AWS les SDK, consultez les [exemples de code](#) dans le manuel Amazon Cognito Developer Guide.

Pour plus de détails sur tous les paramètres du fournisseur d'informations d'identification d'assumer un rôle, consultez [Assumer le rôle de fournisseur d'informations d'identification](#) ce guide.

AWS clés d'accès

Utiliser des identifiants à court terme

Nous vous recommandons de configurer votre SDK ou outil à utiliser [Authentification IAM Identity Center](#) pour utiliser les options de durée de session prolongée.

Toutefois, pour configurer directement les informations d'identification temporaires du SDK ou de l'outil, consultez [Authentifiez-vous à l'aide d'informations d'identification](#).

Utilisez des identifiants à long terme

Warning

Afin d'éviter les risques de sécurité, n'employez pas les utilisateurs IAM pour l'authentification lorsque vous développez des logiciels spécialisés ou lorsque vous travaillez avec des données réelles. Préférez la fédération avec un fournisseur d'identité tel que [AWS IAM Identity Center](#).

Gérez l'accès à travers Comptes AWS

En tant que bonne pratique en matière de sécurité, nous vous recommandons AWS Organizations d'utiliser IAM Identity Center pour gérer l'accès de tous vos Comptes AWS utilisateurs. Pour plus d'informations, consultez la rubrique [Bonnes pratiques IAM](#) du Guide de l'utilisateur IAM.

Vous pouvez créer des utilisateurs dans IAM Identity Center, utiliser Microsoft Active Directory, utiliser un fournisseur d'identité (IdP) SAML 2.0 ou fédérer individuellement votre IdP avec Comptes AWS. En utilisant l'une de ces approches, vous pouvez proposer une expérience d'authentification unique à vos utilisateurs. Vous pouvez également appliquer l'authentification multifactorielle (MFA) et utiliser des informations d'identification Compte AWS temporaires pour l'accès. Cela diffère d'un utilisateur IAM, qui est un identifiant à long terme qui peut être partagé et qui peut augmenter le risque de sécurité pour vos AWS ressources.

Création d'utilisateurs IAM pour les environnements sandbox uniquement

Si vous débutez dans ce domaine AWS, vous pouvez créer un utilisateur IAM de test, puis l'utiliser pour exécuter des didacticiels et découvrir ce que AWS a à offrir. Vous pouvez utiliser ce type d'identifiant lorsque vous apprenez, mais nous vous recommandons d'éviter de l'utiliser en dehors d'un environnement sandbox.

Pour les cas d'utilisation suivants, il peut être judicieux de commencer avec les utilisateurs IAM dans AWS :

- Démarrage avec votre AWS SDK ou outil et exploration Services AWS dans un environnement sandbox.
- Exécution de scripts, de tâches et d'autres processus automatisés planifiés qui ne prennent pas en charge un processus de connexion assisté par un humain dans le cadre de votre apprentissage.

Si vous utilisez des utilisateurs IAM en dehors de ces cas d'utilisation, passez à IAM Identity Center ou fédérez votre fournisseur d'identité Comptes AWS dès que possible. Pour plus d'informations, consultez la section [Fédération des identités dans AWS](#).

Clés d'accès utilisateur IAM sécurisées

Vous devez régulièrement alterner les clés d'accès utilisateur IAM. Suivez les instructions de la section [Rotation des touches d'accès](#) du guide de l'utilisateur IAM. Si vous pensez avoir accidentellement partagé vos clés d'accès utilisateur IAM, faites pivoter vos clés d'accès.

Les clés d'accès utilisateur IAM doivent être stockées dans le `AWS credentials` fichier partagé sur la machine locale. Ne stockez pas les clés d'accès utilisateur IAM dans votre code. N'incluez aucun fichier de configuration contenant vos clés d'accès utilisateur IAM dans un logiciel de gestion de code source. Des outils externes, tels que le projet open source [git-secrets](#), peuvent vous aider à éviter de transférer par inadvertance des informations sensibles dans un dépôt Git. Pour plus d'informations, consultez la section [Identités IAM \(utilisateurs, groupes d'utilisateurs et rôles\)](#) dans le guide de l'utilisateur IAM.

Pour configurer un utilisateur IAM pour qu'il démarre, voir [Authentifiez-vous à l'aide d'informations d'identification](#).

Authentifiez-vous à l'aide d'informations d'identification

Nous vous recommandons de configurer votre SDK ou votre outil pour l'utiliser [Authentification IAM Identity Center](#) avec des options de durée de session prolongée. Toutefois, vous pouvez copier et utiliser les informations d'identification temporaires disponibles sur le portail AWS d'accès. Les nouvelles informations d'identification devront être copiées à leur expiration. Vous pouvez utiliser les informations d'identification temporaires dans un profil ou les utiliser comme valeurs pour les propriétés du système et les variables d'environnement.

Configuration d'un fichier d'informations d'identification à l'aide d'informations d'identification à court terme extraites du portail AWS d'accès

1. [Créez un fichier d'informations d'identification partagé](#).
2. Dans le fichier d'informations d'identification, collez le texte d'espace réservé suivant jusqu'à ce que vous y colliez des informations d'identification temporaires fonctionnelles.

```
[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. Enregistrez le fichier. Le fichier `~/.aws/credentials` devrait maintenant exister sur votre système de développement local. Ce fichier contient le [profil \[par défaut\]](#) utilisé par le SDK ou l'outil si aucun profil nommé spécifique n'est spécifié.
4. [Connectez-vous au portail d'AWSaccès](#).
5. Suivez ces instructions pour [l'actualisation manuelle des informations d'identification](#) afin de copier les informations d'identification du rôle IAM depuis le portail d'AWSaccès.

Avertissements et conseils importants concernant les informations d'identification

Avertissements concernant les informations d'identification

- N'utilisez PAS les informations d'identification root de votre compte pour accéder aux ressources AWS . Ces informations d'identification offrent un accès illimité au compte et sont difficiles à révoquer.
- N'insérez PAS de clés d'accès littérales ou d'informations d'identification dans vos fichiers de candidature. Vous risqueriez en effet d'exposer accidentellement vos informations d'identification si, par exemple, vous chargez le projet sur un référentiel public.
- N'incluez PAS de fichiers contenant des informations d'identification dans votre zone de projet.
- Sachez que toutes les informations d'identification stockées dans le `AWS credentials` fichier partagé sont stockées en texte brut.

Conseils supplémentaires pour gérer les informations d'identification en toute sécurité

Pour une discussion générale sur la manière de gérer les AWS informations d'identification en toute sécurité, consultez la section [Meilleures pratiques pour la gestion des clés AWS d'accès](#) dans le [Références générales AWS](#). En plus de cette discussion, envisagez de prendre les mesures suivantes :

- Utilisez des [rôles IAM](#) pour les tâches Amazon Elastic Container Service (Amazon ECS).
- Utilisez des [rôles IAM](#) pour les applications qui s'exécutent sur des instances Amazon EC2.

Prérequis : créer un compte AWS

Pour utiliser un utilisateur IAM pour accéder aux AWS services, vous avez besoin d'un AWS compte et d' AWS informations d'identification.

1. Créez un compte.

Pour créer un AWS compte, consultez [Commencer : êtes-vous un nouvel AWS utilisateur ?](#) dans le guide AWS Account Management de référence.

2. Créez un utilisateur administratif.

Évitez d'utiliser votre compte utilisateur root (le compte initial que vous créez) pour accéder à la console de gestion et aux services. Au lieu de cela, créez un compte utilisateur administratif, comme décrit à la section [Création d'un utilisateur administratif](#) du Guide de l'utilisateur IAM.

Après avoir créé le compte utilisateur administratif et enregistré les informations de connexion, veuillez à vous déconnecter de votre compte utilisateur root et reconnectez-vous à l'aide du compte administratif.

Aucun de ces comptes n'est approprié pour le développement AWS ou l'exécution d'applications sur AWS. La meilleure pratique consiste à créer des utilisateurs, des ensembles d'autorisations ou des rôles de service adaptés à ces tâches. Pour en savoir plus, consultez [Appliquer les autorisations de moindre privilège](#) dans le Guide de l'utilisateur IAM.


Étape 1 : créer votre utilisateur IAM

- Créez votre utilisateur IAM en suivant la procédure de [création d'utilisateurs IAM \(console\)](#) du guide de l'utilisateur IAM. Lors de la création de votre utilisateur IAM :
 - Nous vous recommandons de sélectionner Fournir un accès utilisateur au AWS Management Console. Cela vous permet d'afficher les Services AWS informations relatives au code que vous exécutez dans un environnement visuel, par exemple en consultant les journaux de AWS CloudTrail diagnostic ou en téléchargeant des fichiers sur Amazon Simple Storage Service, ce qui est utile lors du débogage de votre code.
 - Pour Définir les autorisations - Options d'autorisation, sélectionnez Joindre directement les politiques pour définir la manière dont vous souhaitez attribuer les autorisations à cet utilisateur.
 - La plupart des didacticiels du SDK « Getting Started » utilisent le service Amazon S3 comme exemple. Pour fournir à votre application un accès complet à Amazon S3, sélectionnez la AmazonS3FullAccess politique à associer à cet utilisateur.
 - Vous pouvez ignorer les étapes facultatives de cette procédure concernant la définition des limites d'autorisation ou des balises.

Étape 2 : Obtenez vos clés d'accès

1. Dans le volet de navigation de la console IAM, sélectionnez Utilisateurs, puis sélectionnez **User name** l'utilisateur que vous avez créé précédemment.
2. Sur la page de l'utilisateur, sélectionnez la page Informations d'identification de sécurité. Ensuite, sous Clés d'accès, sélectionnez Créer une clé d'accès.

3. Pour l'étape 1 de création d'une clé d'accès, choisissez soit l'interface de ligne de commande (CLI), soit le code local. Les deux options génèrent le même type de clé à utiliser à la fois avec les SDK AWS CLI et avec les SDK.
4. Pour l'étape 2 de création d'une clé d'accès, entrez une balise facultative et sélectionnez Suivant.
5. Pour l'étape 3 de création d'une clé d'accès, sélectionnez Télécharger le fichier .csv pour enregistrer un .csv fichier avec la clé d'accès et la clé d'accès secrète de votre utilisateur IAM. Vous avez besoin de ces informations pour plus tard.

 Warning

Utilisez les mesures de sécurité appropriées pour protéger ces informations d'identification.

6. Sélectionnez Done (Terminé).

Étape 3 : mettre à jour le **credentials** fichier partagé

1. Créez ou ouvrez le AWS `credentials` fichier partagé. Ce fichier se trouve `~/.aws/credentials` sur les systèmes Linux et macOS, ainsi que `%USERPROFILE%\ .aws \credentials` sur Windows. Pour plus d'informations, voir [Emplacement des fichiers d'informations d'identification](#).
2. Ajoutez le texte suivant au `credentials` fichier partagé. Remplacez l'exemple de valeur d'ID et l'exemple de valeur de clé par les valeurs du .csv fichier que vous avez téléchargé précédemment.

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

3. Enregistrez le fichier.

Le `credentials` fichier partagé est le moyen le plus courant de stocker les informations d'identification. Elles peuvent également être définies en tant que variables d'environnement, voir [AWS clés d'accès](#) pour les noms des variables d'environnement. C'est une façon de démarrer, mais nous vous recommandons de passer à IAM Identity Center ou à d'autres informations d'identification

temporaires dès que possible. Après avoir cessé d'utiliser des informations d'identification à long terme, n'oubliez pas de supprimer ces informations d'identification du `credentials` fichier partagé.

Utilisation de rôles IAM pour les instances Amazon EC2

Cet exemple décrit la configuration d'un AWS Identity and Access Management rôle avec un accès Amazon S3 à utiliser dans votre application déployée sur une instance Amazon EC2.

Pour une instance Amazon Elastic Compute Cloud, créez un rôle IAM, puis accordez à votre instance Amazon EC2 l'accès à ce rôle. Pour plus d'informations, consultez [Rôles IAM pour Amazon EC2](#) dans le guide de l'utilisateur Amazon EC2 ou [Rôles IAM pour Amazon EC2 dans le guide de l'utilisateur Amazon EC2](#).

Créer un rôle IAM

Créez un rôle IAM qui accorde un accès en lecture seule à Amazon S3.

1. Connectez-vous à la console IAM AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/iam/>.
2. Dans le volet de navigation, sélectionnez Rôles, puis sélectionnez Créer un rôle.
3. Pour Sélectionner une entité de confiance, sous Type d'entité fiable, sélectionnez Service AWS.
4. Sous Cas d'utilisation, choisissez Amazon EC2, puis sélectionnez Suivant.
5. Pour Ajouter des autorisations, cochez la case Amazon S3 Read Only Access dans la liste des politiques, puis sélectionnez Suivant.
6. Entrez un nom pour le rôle, puis sélectionnez Créer un rôle. N'oubliez pas ce nom car vous en aurez besoin lorsque vous lancerez votre instance Amazon EC2.

Lancez une instance Amazon EC2 et spécifiez votre rôle IAM

Vous pouvez lancer une instance Amazon EC2 avec un rôle IAM à l'aide de la console Amazon EC2.

Suivez les instructions pour lancer une instance dans le guide de l'utilisateur [Amazon EC2](#) ou le [guide de l'utilisateur Amazon EC2](#).

Lorsque vous atteignez la page Examiner le lancement de l'instance, sélectionnez Modifier les détails de l'instance. Dans Rôle IAM, choisissez le rôle IAM que vous avez créé précédemment. Exécutez la procédure comme indiqué.

Note

Vous devrez créer ou utiliser un groupe de sécurité et une paire de clés existants pour vous connecter à l'instance.

Avec cette configuration IAM et Amazon EC2, vous pouvez déployer votre application sur l'instance Amazon EC2 et elle aura un accès en lecture au service Amazon S3.

Connectez-vous à l'instance EC2

Connectez-vous à l'instance EC2 afin de pouvoir y transférer l'exemple d'application, puis exécuter l'application. Vous aurez besoin du fichier contenant la partie privée de la paire de clés que vous avez utilisée pour lancer l'instance, c'est-à-dire le fichier PEM.

Vous pouvez le faire en suivant la procédure de connexion décrite dans le guide de l'utilisateur [Amazon EC2](#) ou le [guide de l'utilisateur Amazon EC2](#). Lorsque vous vous connectez, faites-le de manière à pouvoir transférer des fichiers de votre machine de développement vers votre instance.

Si vous utilisez un AWS kit d'outils, vous pouvez souvent également vous connecter à l'instance à l'aide du kit d'outils. Pour plus d'informations, consultez le guide de l'utilisateur spécifique à la boîte à outils que vous utilisez.

Exécutez l'exemple d'application sur l'instance EC2

1. Copiez les fichiers d'application de votre disque local vers votre instance.

Pour plus d'informations sur le transfert de fichiers vers votre instance, consultez le guide de l'utilisateur [Amazon EC2](#) ou le [guide de l'utilisateur Amazon EC2](#).

2. Lancez l'application et vérifiez qu'elle s'exécute avec les mêmes résultats que sur votre machine de développement.
3. (Facultatif) Vérifiez que l'application utilise les informations d'identification fournies par le rôle IAM.
 - a. [Connectez-vous à la console Amazon EC2 AWS Management Console et ouvrez-la à l'adresse `https://console.aws.amazon.com/ec2/`](#).
 - b. Sélectionnez l'instance et détachez le rôle IAM via Actions, Paramètres de l'instance, Attacher/remplacer le rôle IAM.

- c. Réexécutez l'application et vérifiez qu'elle renvoie une erreur d'autorisation.

Référence des paramètres

Les SDK fournissent des API spécifiques au langage pour Services AWS. Ils prennent en charge certaines des tâches les plus lourdes nécessaires à la réussite des appels d'API, notamment l'authentification, le comportement des nouvelles tentatives, etc. Pour ce faire, les SDK disposent de stratégies flexibles permettant d'obtenir des informations d'identification à utiliser pour vos demandes, de conserver les paramètres à utiliser avec chaque service et d'obtenir des valeurs à utiliser pour les paramètres globaux.

Vous trouverez des informations détaillées sur les paramètres de configuration dans les sections suivantes :

- [AWS SDK et outils, fournisseurs d'identifiants standardisés](#)— Fournisseurs d'informations d'identification communs normalisés sur plusieurs SDK.
- [AWS Fonctionnalités standardisées des SDK et des outils](#)— Fonctionnalités communes standardisées sur plusieurs SDK.

Création de clients de service

Pour y accéder par programmation Services AWS, les SDK utilisent une classe/un objet client pour chacun d'entre eux. Service AWS Par exemple, si votre application doit accéder à Amazon EC2, elle crée un objet client Amazon EC2 pour établir une interface avec ce service. Vous utilisez ensuite le client du service pour y faire des demandes Service AWS. Dans la plupart des SDK, un objet client de service est immuable. Vous devez donc créer un nouveau client pour chaque service auquel vous envoyez des demandes et pour envoyer des demandes au même service en utilisant une configuration différente.

Priorité des paramètres

Les paramètres globaux configurent les fonctionnalités, les fournisseurs d'informations d'identification et les autres fonctionnalités prises en charge par la plupart des SDK et ayant un large impact sur l'ensemble de ces derniers. Services AWS Tous les SDK comportent une série d'emplacements (ou de sources) qu'ils vérifient afin de trouver une valeur pour les paramètres globaux. La définition de la priorité de recherche est la suivante :

1. Tout paramètre explicite défini dans le code ou sur un client de service lui-même a priorité sur tout autre paramètre.

- Certains paramètres peuvent être définis pour chaque opération et peuvent être modifiés selon les besoins pour chaque opération que vous invoquez. Pour le AWS CLI ou AWS Tools for PowerShell, ils prennent la forme de paramètres par opération que vous entrez sur la ligne de commande. Pour un SDK, les attributions explicites peuvent prendre la forme d'un paramètre que vous définissez lorsque vous instanciez un Service AWS client ou un objet de configuration, ou parfois lorsque vous appelez une API individuelle.
2. Java/Kotlin uniquement : la propriété du système JVM pour le paramètre est vérifiée. Si elle est définie, cette valeur est utilisée pour configurer le client.
 3. La variable d'environnement est contrôlée. Si elle est définie, cette valeur est utilisée pour configurer le client.
 4. Le SDK vérifie le paramètre dans `credentials` le fichier partagé. S'il est défini, le client l'utilise.
 5. Le `config` fichier partagé pour le paramètre. Si le paramètre est présent, le SDK l'utilise.
 - La variable d'`AWS_PROFILE` environnement ou la propriété du système `aws.profile` JVM peuvent être utilisées pour spécifier le profil chargé par le SDK.
 6. Toute valeur par défaut fournie par le code source du SDK lui-même est utilisée en dernier.

Note

Certains SDK et outils peuvent être enregistrés dans un ordre différent. En outre, certains SDK et outils prennent en charge d'autres méthodes de stockage et de récupération de paramètres. Par exemple, il AWS SDK for .NET prend en charge une source supplémentaire appelée [SDK Store](#). Pour plus d'informations sur les fournisseurs spécifiques à un SDK ou à un outil, consultez le guide spécifique au SDK ou à l'outil que vous utilisez.

L'ordre détermine quelles méthodes ont priorité et remplacent les autres. Par exemple, si vous configurez un profil dans le `config` fichier partagé, il n'est trouvé et utilisé qu'une fois que le SDK ou l'outil a d'abord vérifié les autres emplacements. Cela signifie que si vous insérez un paramètre dans le `credentials` fichier, il est utilisé à la place de celui qui se trouve dans le `config` fichier. Si vous configurez une variable d'environnement avec un paramètre et une valeur, elle remplacera ce paramètre dans les `config` fichiers `credentials` et. Enfin, un réglage sur l'opération individuelle (paramètre de AWS CLI ligne de commande ou paramètre d'API) ou dans le code remplacerait toutes les autres valeurs de cette commande.

Configliste des paramètres de fichier

Les paramètres répertoriés dans le tableau suivant peuvent être attribués dans le AWS config fichier partagé. Elles sont mondiales et concernent tout le monde Services AWS. Les SDK et les outils peuvent également prendre en charge des paramètres et des variables d'environnement uniques. Pour voir les paramètres et les variables d'environnement pris en charge uniquement par un SDK ou un outil individuel, consultez ce SDK ou ce guide d'outils spécifique.

Nom du paramètre	Détails
api_versions	Paramètres de configuration généraux
aws_access_key_id	AWS clés d'accès
aws_secret_access_key	AWS clés d'accès
aws_session_token	AWS clés d'accès
ca_bundle	Paramètres de configuration généraux
credential_process	Fournisseur d'identifiants de processus
credential_source	Assumer le rôle de fournisseur d'informations d'identification
defaults_mode	Paramètres de configuration intelligents par défaut
disable_request_compression	Compression des demandes
duration_seconds	Assumer le rôle de fournisseur d'informations d'identification

Nom du paramètre	Détails
ec2_metadata_service_endpoint	fournisseur d'informations d'identification IMDS
ec2_metadata_service_endpoint_mode	fournisseur d'informations d'identification IMDS
ec2_metadata_v1_disabled	fournisseur d'informations d'identification IMDS
endpoint_discovery_enabled	Découverte des terminaux
endpoint_url	Points de terminaison spécifiques au service
external_id	Assumer le rôle de fournisseur d'informations d'identification
ignore_configured_endpoint_urls	Points de terminaison spécifiques au service
max_attempts	Comportement des nouvelles tentatives
metadata_service_num_attempts	Métadonnées de l'instance Amazon EC2
metadata_service_timeout	Métadonnées de l'instance Amazon EC2

Nom du paramètre	Détails
mfa_serial	Assumer le rôle de fournisseur d'informations d'identification
output	Paramètres de configuration généraux
parameter_validation	Paramètres de configuration généraux
region	Région AWS
request_min_compression_size_bytes	Compression des demandes
retry_mode	Comportement des nouvelles tentatives
role_arn	Assumer le rôle de fournisseur d'informations d'identification
role_session_name	Assumer le rôle de fournisseur d'informations d'identification
s3_disable_multiregion_access_points	Amazon S3 Multi-Region Access Points
s3_use_arn_region	Points d'accès Amazon S3
sdk_ua_app_id	ID d'application
source_profile	Assumer le rôle de fournisseur d'informations d'identification
sso_account_id	Fournisseur d'identifiants IAM Identity Center

Nom du paramètre	Détails
<code>sso_region</code>	Fournisseur d'identifiants IAM Identity Center
<code>sso_registration_scopes</code>	Fournisseur d'identifiants IAM Identity Center
<code>sso_role_name</code>	Fournisseur d'identifiants IAM Identity Center
<code>sso_start_url</code>	Fournisseur d'identifiants IAM Identity Center
<code>sts_regional_endpoints</code>	AWS STS Points de terminaison régionalisés
<code>use_dualstack_endpoint</code>	Points de terminaison à double pile et FIPS
<code>use_fips_endpoint</code>	Points de terminaison à double pile et FIPS
<code>web_identity_token_file</code>	Assumer le rôle de fournisseur d'informations d'identification

Credentials liste des paramètres de fichier

Les paramètres répertoriés dans le tableau suivant peuvent être attribués dans le AWS `credentials` fichier partagé. Elles sont mondiales et concernent tout le monde Services AWS. Les SDK et les outils peuvent également prendre en charge des paramètres et des variables d'environnement uniques. Pour voir les paramètres et les variables d'environnement pris en charge uniquement par un SDK ou un outil individuel, consultez ce SDK ou ce guide d'outils spécifique.

Nom du paramètre	Détails
<code>aws_access_key_id</code>	AWS clés d'accès
<code>aws_secret_access_key</code>	AWS clés d'accès

Nom du paramètre	Détails
aws_session_token	AWS clés d'accès

Liste des variables d'environnement

Les variables d'environnement prises en charge par la plupart des SDK sont répertoriées dans le tableau suivant. Elles sont mondiales et concernent tout le monde Services AWS. Les SDK et les outils peuvent également prendre en charge des paramètres et des variables d'environnement uniques. Pour voir les paramètres et les variables d'environnement pris en charge uniquement par un SDK ou un outil individuel, consultez ce SDK ou ce guide d'outils spécifique.

Nom du paramètre	Détails
AWS_ACCESS_KEY_ID	AWS clés d'accès
AWS_CA_BUNDLE	Paramètres de configuration généraux
AWS_CONFIG_FILE	Emplacement du partage config et des credentials fichiers
AWS_CONTAINER_AUTHORIZATION_TOKEN	Fournisseur d'informations d'identification du conteneur
AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE	Fournisseur d'informations d'identification du conteneur
AWS_CONTAINER_CREDENTIALS_FULL_URI	Fournisseur d'informations d'identification du conteneur

Nom du paramètre	Détails
AWS_CONTA INER_CRED ENTIALS_R ELATIVE_URI	Fournisseur d'informations d'identification du conteneur
AWS_DEFAU LTS_MODE	Paramètres de configuration intelligents par défaut
AWS_DISAB LE_REQUES T_COMPRESSION	Compression des demandes
AWS_EC2_M ETADATA_D ISABLED	fournisseur d'informations d'identification IMDS
AWS_EC2_M ETADATA_S ERVICE_EN DPOINT	fournisseur d'informations d'identification IMDS
AWS_EC2_M ETADATA_S ERVICE_EN DPOINT_MODE	fournisseur d'informations d'identification IMDS
AWS_EC2_M ETADATA_V 1_DISABLED	fournisseur d'informations d'identification IMDS
AWS_ENABL E_ENDPOIN T_DISCOVERY	Découverte des terminaux
AWS_ENDPO INT_URL	Points de terminaison spécifiques au service

Nom du paramètre	Détails
AWS_ENDPOINT_URL_<SERVICE>	Points de terminaison spécifiques au service
AWS_IAM_ROLE_ARN	Assumer le rôle de fournisseur d'informations d'identification
AWS_IAM_ROLE_SESSION_NAME	Assumer le rôle de fournisseur d'informations d'identification
AWS_IGNORE_ENDPOINT_URLS	Points de terminaison spécifiques au service
AWS_MAX_ATTEMPTS	Comportement des nouvelles tentatives
AWS_METADATA_SERVICE_NUM_ATTEMPTS	Métadonnées de l'instance Amazon EC2
AWS_METADATA_SERVICE_TIMEOUT	Métadonnées de l'instance Amazon EC2
AWS_PROFILE	Partage config et credentials fichiers
AWS_REGION	Région AWS
AWS_REQUEST_COMPRESSION_SIZE_BYTES	Compression des demandes

Nom du paramètre	Détails
AWS_RETRY_MODE	Comportement des nouvelles tentatives
AWS_S3_DISABLE_MULTIREGION_ACCESS_POINTS	Amazon S3 Multi-Region Access Points
AWS_S3_USE_ARN_REGION	Points d'accès Amazon S3
AWS_SDK_USER_APP_ID	ID d'application
AWS_SECRET_ACCESS_KEY	AWS clés d'accès
AWS_SESSION_TOKEN	AWS clés d'accès
AWS_SHARED_CREDENTIALS_FILE	Emplacement du partage config et des credentials fichiers
AWS_STS_REGIONAL_ENDPOINTS	AWS STS Points de terminaison régionalisés
AWS_USE_DUALSTACK_ENDPOINT	Points de terminaison à double pile et FIPS
AWS_USE_FIPS_ENDPOINT	Points de terminaison à double pile et FIPS
AWS_WEB_IDENTITY_TOKEN_FILE	Assumer le rôle de fournisseur d'informations d'identification

Liste des propriétés du système JVM

Vous pouvez utiliser les propriétés du système JVM suivantes pour le AWS SDK for Java et Kit AWS SDK pour Kotlin (en ciblant la JVM). Consultez [the section called “Comment définir les propriétés du système JVM”](#) les instructions sur la façon de définir les propriétés du système JVM.

Nom du paramètre	Détails
<code>aws.accessKeyId</code>	AWS clés d'accès
<code>aws.configFile</code>	Emplacement du partage config et des credentials fichiers
<code>aws.defaultsMode</code>	Paramètres de configuration intelligents par défaut
<code>aws.disableEc2MetadataV1</code>	fournisseur d'informations d'identification IMDS
<code>aws.disableRequestCompression</code>	Compression des demandes
<code>aws.ec2MetadataServiceEndpoint</code>	fournisseur d'informations d'identification IMDS
<code>aws.ec2MetadataServiceEndpointMode</code>	fournisseur d'informations d'identification IMDS
<code>aws.endpointDiscoveryEnabled</code>	Découverte des terminaux
<code>aws.endpointUrl</code>	Points de terminaison spécifiques au service

Nom du paramètre	Détails
<code>aws.endpointUrl<ServiceName></code>	Points de terminaison spécifiques au service
<code>aws.ignoreConfiguredEndpointUrls</code>	Points de terminaison spécifiques au service
<code>aws.maxAttempts</code>	Comportement des nouvelles tentatives
<code>aws.profile</code>	Partage config et credentials fichiers
<code>aws.region</code>	Région AWS
<code>aws.requestMinCompressionSizeBytes</code>	Compression des demandes
<code>aws.retryMode</code>	Comportement des nouvelles tentatives
<code>aws.roleArn</code>	Assumer le rôle de fournisseur d'informations d'identification
<code>aws.roleSessionName</code>	Assumer le rôle de fournisseur d'informations d'identification
<code>aws.s3DisableMultiRegionAccessPoints</code>	Amazon S3 Multi-Region Access Points
<code>aws.s3UseArnRegion</code>	Points d'accès Amazon S3
<code>aws.secretAccessKey</code>	AWS clés d'accès

Nom du paramètre	Détails
<code>aws.sessionToken</code>	AWS clés d'accès
<code>aws.sharedCredentialsFile</code>	Emplacement du partage config et des credentials fichiers
<code>aws.useDualstackEndpoint</code>	Points de terminaison à double pile et FIPS
<code>aws.useFipsEndpoint</code>	Points de terminaison à double pile et FIPS
<code>aws.userAgentAppId</code>	ID d'application
<code>aws.webIdentityTokenFile</code>	Assumer le rôle de fournisseur d'informations d'identification

AWS SDK et outils, fournisseurs d'identifiants standardisés

De nombreux fournisseurs d'identifiants ont été standardisés selon des valeurs par défaut cohérentes et fonctionnent de la même manière sur de nombreux SDK. Cette cohérence augmente la productivité et la clarté lors du codage sur plusieurs SDK. Tous les paramètres peuvent être remplacés dans le code. Pour plus de détails, consultez l'API de votre SDK spécifique.

Important

Tous les SDK ne prennent pas en charge tous les fournisseurs, ni même tous les aspects d'un fournisseur.

Rubriques

- [Chaîne de fournisseurs d'identifiants](#)

- [AWS clés d'accès](#)
- [Assumer le rôle de fournisseur d'informations d'identification](#)
- [Fournisseur d'informations d'identification du conteneur](#)
- [Fournisseur d'identifiants IAM Identity Center](#)
- [fournisseur d'informations d'identification IMDS](#)
- [Fournisseur d'identifiants de processus](#)

Chaîne de fournisseurs d'identifiants

Tous les SDK ont une série d'emplacements (ou de sources) qu'ils vérifient afin de trouver des informations d'identification valides à utiliser pour envoyer une demande à un Service AWS. Une fois les informations d'identification valides trouvées, la recherche s'arrête. Cette recherche systématique est appelée chaîne de fournisseurs d'informations d'identification par défaut.

Bien que la chaîne distincte utilisée par chaque SDK varie, ils incluent le plus souvent des sources telles que les suivantes :

Fournisseur d'identifiants	Description
AWS clés d'accès	AWS clés d'accès pour un utilisateur IAM (telles que <code>AWS_ACCESS_KEY_ID</code> , et <code>AWS_SECRET_ACCESS_KEY</code>).
Fédérez avec l'identité Web ou OpenID Connect - Assumer le rôle de fournisseur d'informations d'identification	Connectez-vous à l'aide d'un fournisseur d'identité externe (IdP) connu, tel que Login with Amazon, Facebook, Google ou tout autre IdP compatible avec OpenID Connect (OIDC). Assumez les autorisations d'un rôle IAM à l'aide d'un jeton d'identité Web provenant de AWS Security Token Service (AWS STS).
Fournisseur d'identifiants IAM Identity Center	Obtenez des informations d'identification auprès de AWS IAM Identity Center.
Assumer le rôle de fournisseur d'informations d'identification	Accédez à d'autres ressources en assumant les autorisations d'un rôle IAM. (Récupérez puis utilisez les informations d'identification temporaires pour un rôle).

Fournisseur d'identifiants	Description
Fournisseur d'informations d'identification du conteneur	Informations d'identification Amazon Elastic Container Service (Amazon ECS) et Amazon Elastic Kubernetes Service (Amazon EKS). Le fournisseur d'informations d'identification du conteneur récupère les informations d'identification de l'application conteneurisée du client.
Fournisseur d'identifiants de processus	Fournisseur d'informations d'identification personnalisé. Obtenez vos informations d'identification auprès d'une source ou d'un processus externe, notamment IAM Roles Anywhere.
fournisseur d'informations d'identification IMDS	Informations d'identification du profil d'instance Amazon Elastic Compute Cloud (Amazon EC2). Associez un rôle IAM à chacune de vos instances EC2. Les informations d'identification temporaires pour ce rôle sont mises à la disposition du code exécuté dans l'instance. Les informations d'identification sont fournies via le service de métadonnées Amazon EC2.

Pour chaque étape de la chaîne, il existe plusieurs manières d'attribuer des valeurs de réglage. La définition des valeurs spécifiées dans le code est toujours prioritaire. Cependant, il y a aussi [Variables d'environnement](#) et les [Partage config et credentials fichiers](#). Pour plus d'informations, voir [Priorité des paramètres](#).

AWS clés d'accès

Warning

Afin d'éviter les risques de sécurité, n'employez pas les utilisateurs IAM pour l'authentification lorsque vous développez des logiciels spécialisés ou lorsque vous travaillez avec des données réelles. Préférez la fédération avec un fournisseur d'identité tel que [AWS IAM Identity Center](#).

AWS les clés d'accès d'un utilisateur IAM peuvent être utilisées comme AWS informations d'identification. Le AWS SDK utilise automatiquement ces AWS informations d'identification pour signer les demandes d'API AWS, afin que vos charges de travail puissent accéder à vos AWS ressources et à vos données de manière sûre et pratique. Il est recommandé de toujours utiliser le `aws_session_token` afin que les informations d'identification soient temporaires et ne soient plus valides après leur expiration. L'utilisation d'informations d'identification à long terme n'est pas recommandée.

Note

Si vous AWS ne parvenez pas à actualiser ces informations d'identification temporaires, cela AWS peut prolonger la validité des informations d'identification afin que vos charges de travail ne soient pas affectées.

Le AWS `credentials` fichier partagé est l'emplacement recommandé pour stocker les informations d'identification, car il se trouve en toute sécurité en dehors des répertoires des sources de l'application et distinct des paramètres spécifiques au SDK du fichier partagé `config`.

Pour en savoir plus sur les informations AWS d'identification et l'utilisation des clés d'accès, consultez les [AWS sections Identifiants de sécurité](#) et [Gestion des clés d'accès pour les utilisateurs IAM](#) dans le Guide de l'utilisateur IAM.

Configurez cette fonctionnalité à l'aide des méthodes suivantes :

`aws_access_key_id`- réglage AWS `config` du fichier partagé, **`aws_access_key_id`**- réglage AWS `credentials` du fichier partagé (méthode recommandée), **`AWS_ACCESS_KEY_ID`**- variable d'environnement, **`aws.accessKeyId`**- Propriété du système JVM : Java/Kotlin uniquement

Spécifie la clé AWS d'accès utilisée dans le cadre des informations d'identification pour authentifier l'utilisateur.

`aws_secret_access_key`- réglage AWS `config` du fichier partagé, **`aws_secret_access_key`**- réglage AWS `credentials` du fichier partagé (méthode recommandée), **`AWS_SECRET_ACCESS_KEY`**- variable d'environnement, **`aws.secretAccessKey`**- Propriété du système JVM : Java/Kotlin uniquement

Spécifie la clé AWS secrète utilisée dans le cadre des informations d'identification pour authentifier l'utilisateur.

aws_session_token- réglage **aws config** du fichier partagé, **aws_session_token**- réglage **AWS credentials** du fichier partagé (méthode recommandée), **AWS_SESSION_TOKEN**- variable d'environnement, **aws.sessionToken**- Propriété du système JVM : Java/Kotlin uniquement

Spécifie un jeton de AWS session utilisé dans le cadre des informations d'identification pour authentifier l'utilisateur. Vous recevez cette valeur dans le cadre des informations d'identification temporaires renvoyées en cas de réussite des demandes d'attribution d'un rôle. Un jeton de session n'est nécessaire que si vous spécifiez manuellement des informations d'identification de sécurité temporaires. Toutefois, nous vous recommandons de toujours utiliser des informations d'identification de sécurité temporaires plutôt que des informations d'identification à long terme. Pour les recommandations en matière de sécurité, consultez [la section Meilleures pratiques en matière de sécurité dans IAM](#).

Pour obtenir des instructions sur la façon d'obtenir ces valeurs, reportez-vous à [Authentifiez-vous à l'aide d'informations d'identification](#).

Exemple de définition de ces valeurs obligatoires dans le `credentials` fichier `config` or :

```
[default]
aws_access_key_id = AKIAIOSFODNN7EXAMPLE
aws_secret_access_key = wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY
aws_session_token = AQoEXAMPLEH4aoAH0gNCAPy...truncated...zrkuWJ0gQs8IZZaIv2BXIa2R40Lgk
```

Exemple Linux/macOS de définition de variables d'environnement via la ligne de commande :

```
export AWS_ACCESS_KEY_ID=AKIAIOSFODNN7EXAMPLE
export AWS_SECRET_ACCESS_KEY=wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY
export
AWS_SESSION_TOKEN=AQoEXAMPLEH4aoAH0gNCAPy...truncated...zrkuWJ0gQs8IZZaIv2BXIa2R40Lgk
```

Exemple Windows de définition de variables d'environnement via la ligne de commande :

```
setx AWS_ACCESS_KEY_ID AKIAIOSFODNN7EXAMPLE
setx AWS_SECRET_ACCESS_KEY wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY
setx
AWS_SESSION_TOKEN AQoEXAMPLEH4aoAH0gNCAPy...truncated...zrkuWJ0gQs8IZZaIv2BXIa2R40Lgk
```

Compatibilité avec les AWS SDK

Les SDK suivants prennent en charge les fonctionnalités et les paramètres décrits dans cette rubrique. Toute exception partielle est notée. Tous les paramètres de propriété du système JVM sont pris en charge par le AWS SDK for Java et le Kit AWS SDK pour Kotlin seul.

SDK	Pr er cf	Remarques ou informations supplémentaires
AWS CLI v2	Oui	
SDK pour C++	Oui	config file partagé n'est pas pris en charge.
SDK pour Go V2 (1.x)	Oui	
SDK pour Go 1.x (V1)	Oui	Pour utiliser les paramètres des config fichiers partagés, vous devez activer le chargement à partir du fichier de configuration ; voir Sessions .
SDK pour Java 2.x	Oui	
SDK pour Java 1.x	Oui	
SDK pour 3.x JavaScript	Oui	
SDK pour 2.x JavaScript	Oui	
SDK pour Kotlin	Oui	
SDK pour .NET 3.x	Oui	Les variables d'environnement ne sont pas prises en charge.
SDK pour PHP 3.x	Oui	
SDK pour Python (Boto3)	Oui	
SDK pour Ruby 3.x	Oui	
SDK pour Rust	Oui	
Outils pour PowerShell	Oui	Les variables d'environnement ne sont pas prises en charge.

Assumer le rôle de fournisseur d'informations d'identification

Assumer un rôle implique l'utilisation d'un ensemble d'informations d'identification de sécurité temporaires pour accéder à AWS des ressources auxquelles vous n'auriez peut-être pas accès autrement. Ces informations d'identification temporaires incluent un ID de clé d'accès, une clé d'accès secrète et un jeton de sécurité.

Pour configurer votre SDK ou votre outil afin qu'il assume un rôle, vous devez d'abord créer ou identifier un rôle spécifique à assumer. Les rôles IAM sont identifiés de manière unique par un rôle Amazon Resource Name ([ARN](#)). Les rôles établissent des relations de confiance avec une autre entité. L'entité de confiance qui utilise le rôle peut être un Service AWS fournisseur d'identité Web Compte AWS, une fédération OIDC ou SAML.

Une fois le rôle IAM identifié, si ce rôle vous fait confiance, vous pouvez configurer votre SDK ou votre outil pour utiliser les autorisations accordées par le rôle. Pour ce faire, utilisez les paramètres suivants.

Pour savoir comment commencer à utiliser ces paramètres, consultez [Assumer un rôle](#) ce guide.

Assumer les paramètres du fournisseur d'informations d'identification du rôle

Configurez cette fonctionnalité à l'aide des méthodes suivantes :

credential_source- réglage AWS **config** du fichier partagé

Utilisé dans les instances Amazon EC2 ou les conteneurs Amazon Elastic Container Service pour spécifier où le SDK ou l'outil peut trouver les informations d'identification autorisées à assumer le rôle que vous spécifiez avec le paramètre. `role_arn`

Valeur par défaut : Aucune

Valeurs valides:

- Environnement — Spécifie que le SDK ou l'outil doit récupérer les informations d'identification de la source à partir des variables d'environnement [AWS_ACCESS_KEY_ID](#) et [AWS_SECRET_ACCESS_KEY](#).
- Ec2 InstanceMetadata — Spécifie que le SDK ou l'outil doit utiliser le [rôle IAM attaché au profil d'instance EC2 pour obtenir les informations](#) d'identification de la source.
- EcsContainer— Spécifie que le SDK ou l'outil doit utiliser le [rôle IAM attaché au conteneur ECS pour obtenir les informations d'identification](#) de la source.

Vous ne pouvez pas spécifier à la fois `credential_source` et `source_profile` dans le même profil.

Exemple de définition de ce paramètre dans un config fichier pour indiquer que les informations d'identification doivent provenir d'Amazon EC2 :

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

duration_seconds- réglage AWS **config** du fichier partagé

Spécifie la durée maximale de la session de rôle, en secondes.

Ce paramètre s'applique uniquement lorsque le profil indique d'assumer un rôle.

Valeur par défaut : 3 600 secondes (une heure)

Valeurs valides : La valeur peut aller de 900 secondes (15 minutes) à la durée maximale de session configurée pour le rôle (qui peut être de 43 200 secondes ou 12 heures au maximum). Pour plus d'informations, voir [Afficher le paramètre de durée maximale de session pour un rôle](#) dans le guide de l'utilisateur IAM.

Exemple de configuration de ce paramètre dans un config fichier :

```
duration_seconds = 43200
```

external_id- réglage AWS **config** du fichier partagé

Identifiant unique utilisé par des tiers pour assumer un rôle dans les comptes de leurs clients.

Ce paramètre s'applique uniquement lorsque le profil indique d'assumer un rôle et que la politique de confiance associée à ce rôle nécessite une valeur pour `ExternalId`. La valeur correspond au `ExternalId` paramètre transmis à l'`AssumeRole` opération lorsque le profil spécifie un rôle.

Valeur par défaut : Aucune.

Valeurs valides : consultez la section [Comment utiliser un identifiant externe lorsque vous accordez l'accès à vos AWS ressources à un tiers](#) dans le guide de l'utilisateur IAM.

Exemple de configuration de ce paramètre dans un config fichier :

```
external_id = unique_value_assigned_by_3rd_party
```

mfa_serial- réglage AWS **config** du fichier partagé

Spécifie l'identification ou le numéro de série d'un dispositif d'authentification multifactorielle (MFA) que l'utilisateur doit utiliser lorsqu'il assume un rôle.

Obligatoire lorsque vous assumez un rôle où la politique de confiance associée à ce rôle inclut une condition nécessitant une authentification MFA.

Valeur par défaut : Aucune.

Valeurs valides : la valeur peut être soit un numéro de série pour un périphérique matériel (tel que GAHT12345678), soit un Amazon Resource Name (ARN) pour un périphérique MFA virtuel. Pour plus d'informations sur le MFA, consultez la [section Configuration de l'accès aux API protégé par MFA](#) dans le guide de l'utilisateur IAM.

Exemple de configuration de ce paramètre dans un config fichier :

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

role_arn- réglage AWS **config** du fichier partagé, **AWS_IAM_ROLE_ARN**- variable d'environnement, **aws.roleArn**- Propriété du système JVM : Java/Kotlin uniquement

Spécifie le nom de ressource Amazon (ARN) d'un rôle IAM que vous souhaitez utiliser pour effectuer les opérations demandées à l'aide de ce profil.

Valeur par défaut : Aucune.

Valeurs valides : la valeur doit être l'ARN d'un rôle IAM, formaté comme suit :

```
arn:aws:iam::account-id:role/role-name
```

En outre, vous devez également définir l'un des paramètres suivants :

- **source_profile**— Pour identifier un autre profil à utiliser pour trouver les informations d'identification autorisées à assumer le rôle dans ce profil.
- **credential_source**— Pour utiliser les informations d'identification identifiées par les variables d'environnement actuelles ou les informations d'identification associées à un profil d'instance Amazon EC2 ou à une instance de conteneur Amazon ECS.

- `web_identity_token_file`— Utiliser des fournisseurs d'identité publics ou tout autre fournisseur d'identité compatible avec OpenID Connect (OIDC) pour les utilisateurs authentifiés dans une application mobile ou Web.

role_session_name- réglage AWS **config** du fichier partagé, **AWS_IAM_ROLE_SESSION_NAME**-variable d'environnement, **aws.roleSessionName**- Propriété du système JVM : Java/Kotlin uniquement

Spécifie le nom à attacher à la session de rôle. Ce nom apparaît dans AWS CloudTrail les journaux des entrées associées à cette session, ce qui peut être utile lors d'un audit.

Valeur par défaut : paramètre facultatif. Si vous ne fournissez pas cette valeur, un nom de session est généré automatiquement si le profil assume un rôle.

Valeurs valides : fournies au `RoleSessionName` paramètre lorsque l' AWS API AWS CLI or appelle l'`AssumeRole`opération (ou des opérations telles que l'`AssumeRoleWithWebIdentity`opération) en votre nom. La valeur fait partie du rôle assumé par l'utilisateur Amazon Resource Name (ARN) que vous pouvez interroger et apparaît dans les entrées du CloudTrail journal pour les opérations invoquées par ce profil.

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

Exemple de configuration de ce paramètre dans un config fichier :

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

source_profile- réglage AWS **config** du fichier partagé

Spécifie un autre profil dont les informations d'identification sont utilisées pour assumer le rôle spécifié par le `role_arn` paramètre du profil d'origine. Pour comprendre comment les profils sont utilisés dans le partage AWS config et `credentials` les fichiers, consultez [Partage config et credentials fichiers](#).

Si vous spécifiez un profil qui est également un profil d'acceptation de rôle, chaque rôle sera assumé dans un ordre séquentiel afin de résoudre complètement les informations d'identification. Cette chaîne est arrêtée lorsque le SDK rencontre un profil avec des informations d'identification. Le chaînage des rôles limite votre session de rôle AWS CLI ou celle de l' AWS API à une heure maximum et ne peut pas être augmenté. Pour plus d'informations, consultez la section [Termes et concepts relatifs aux rôles](#) dans le guide de l'utilisateur IAM.

Valeur par défaut : Aucune.

Valeurs valides : cette valeur doit être un chemin et un nom de fichier. Le fichier doit contenir un jeton d'accès OAuth 2.0 ou un jeton OpenID Connect qui vous a été fourni par un fournisseur d'identité. Les chemins relatifs sont traités comme relatifs au répertoire de travail du processus.

Compatibilité avec les AWS SDK

Les SDK suivants prennent en charge les fonctionnalités et les paramètres décrits dans cette rubrique. Toute exception partielle est notée. Tous les paramètres de propriété du système JVM sont pris en charge par le AWS SDK for Java et le Kit AWS SDK pour Kotlin seul.

SDK	Partie	Remarques ou informations supplémentaires
AWS CLI v2	Oui	
SDK pour C++	Partie	<code>credential_source</code> non pris en charge. <code>duration_seconds</code> non pris en charge. <code>mfa_serial</code> non pris en charge.
SDK pour Go V2 (1.x)	Oui	
SDK pour Go 1.x (V1)	Oui	Pour utiliser les paramètres des config fichiers partagés, vous devez activer le chargement à partir du fichier de configuration ; voir Sessions .
SDK pour Java 2.x	Partie	<code>mfa_serial</code> non pris en charge. Utiliser <code>AWS_ROLE_ARN</code> au lieu de <code>AWS_IAM_ROLE_ARN</code> . Utiliser <code>AWS_ROLE_SESSION_NAME</code> au lieu de <code>AWS_IAM_ROLE_SESSION_NAME</code> .
SDK pour Java 1.x	Partie	<code>mfa_serial</code> non pris en charge. Les propriétés du système JVM ne sont pas prises en charge.
SDK pour 3.x JavaScript	Oui	
SDK pour 2.x JavaScript	Partie	<code>credential_source</code> non pris en charge.

SDK	Pré-requis	Remarques ou informations supplémentaires
SDK pour Kotlin	Oui	Utiliser <code>AWS_ROLE_ARN</code> au lieu de <code>AWS_IAM_ROLE_ARN</code> . Utiliser <code>AWS_ROLE_SESSION_NAME</code> au lieu de <code>AWS_IAM_ROLE_SESSION_NAME</code> .
SDK pour .NET 3.x	Oui	
SDK pour PHP 3.x	Oui	
SDK pour Python (Boto3)	Oui	
SDK pour Ruby 3.x	Oui	
SDK pour Rust	Oui	
Outils pour PowerShell	Oui	

Fournisseur d'informations d'identification du conteneur

Le fournisseur d'informations d'identification du conteneur récupère les informations d'identification de l'application conteneurisée du client. Ce fournisseur d'informations d'identification est utile pour les clients d'Amazon Elastic Container Service (Amazon ECS) et d'Amazon Elastic Kubernetes Service (Amazon EKS). Les SDK tentent de charger les informations d'identification depuis le point de terminaison HTTP spécifié via une requête GET.

Si vous utilisez Amazon ECS, nous vous recommandons d'utiliser un rôle IAM de tâche pour améliorer l'isolation, l'autorisation et l'auditabilité des informations d'identification. Une fois configuré, Amazon ECS définit la variable d'environnement `AWS_CONTAINER_CREDENTIALS_RELATIVE_URI` que les SDK et les outils utilisent pour obtenir des informations d'identification. Pour configurer Amazon ECS pour cette fonctionnalité, consultez le [rôle de Task IAM](#) dans le manuel Amazon Elastic Container Service Developer Guide.

Si vous utilisez Amazon EKS, nous vous recommandons d'utiliser Amazon EKS Pod Identity pour améliorer l'isolation des informations d'identification, le moindre privilège, l'auditabilité, le fonctionnement indépendant, la réutilisabilité et l'évolutivité. Votre Pod et un rôle IAM sont associés à un compte de service Kubernetes pour gérer les informations d'identification de

vos applications. Pour en savoir plus sur Amazon EKS Pod Identity, consultez [Amazon EKS Pod Identities](#) dans le guide de l'utilisateur Amazon EKS. Une fois configuré, Amazon EKS définit les variables d'AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE d'environnement AWS_CONTAINER_CREDENTIALS_FULL_URI et d'environnement utilisées par les SDK et les outils pour obtenir des informations d'identification. Pour plus d'informations sur la configuration, consultez [Configuration de l'agent Amazon EKS Pod Identity](#) dans le guide de l'utilisateur Amazon EKS ou [Amazon EKS Pod Identity simplifie les autorisations IAM pour les applications sur les clusters Amazon EKS](#) sur le site Web du AWS blog.

Configurez cette fonctionnalité à l'aide des méthodes suivantes :

AWS_CONTAINER_CREDENTIALS_FULL_URI- variable d'environnement

Spécifie le point de terminaison de l'URL HTTP complet que le SDK doit utiliser lors d'une demande d'informations d'identification. Cela inclut à la fois le schéma et l'hôte.

Valeur par défaut : Aucune.

Valeurs valides : URI valide.

Remarque : Ce paramètre est une alternative *AWS_CONTAINER_CREDENTIALS_RELATIVE_URI* et ne sera utilisé que s'il n'*AWS_CONTAINER_CREDENTIALS_RELATIVE_URI* est pas défini.

Exemple Linux/macOS de définition de variables d'environnement via la ligne de commande :

```
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- variable d'environnement

Spécifie le point de terminaison de l'URL HTTP relative que le SDK doit utiliser lors d'une demande d'informations d'identification. La valeur est ajoutée au nom d'hôte Amazon ECS par défaut de. 169.254.170.2

Valeur par défaut : Aucune.

Valeurs valides : URI relative valide.

Exemple Linux/macOS de définition de variables d'environnement via la ligne de commande :

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

AWS_CONTAINER_AUTHORIZATION_TOKEN- variable d'environnement

Spécifie un jeton d'autorisation en texte brut. Si cette variable est définie, le SDK définira l'en-tête d'autorisation de la requête HTTP avec la valeur de la variable d'environnement.

Valeur par défaut : Aucune.

Valeurs valides : chaîne.

Remarque : Ce paramètre est une alternative `AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE` et ne sera utilisé que s'il n'`AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE` est pas défini.

Exemple Linux/macOS de définition de variables d'environnement via la ligne de commande :

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

AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE- variable d'environnement

Spécifie un chemin de fichier absolu vers un fichier contenant le jeton d'autorisation en texte brut.

Valeur par défaut : Aucune.

Valeurs valides : chaîne.

Exemple Linux/macOS de définition de variables d'environnement via la ligne de commande :

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

Compatibilité avec les AWS SDK

Les SDK suivants prennent en charge les fonctionnalités et les paramètres décrits dans cette rubrique. Toute exception partielle est notée. Tous les paramètres de propriété du système JVM sont pris en charge par le AWS SDK for Java et le Kit AWS SDK pour Kotlin seul.

SDK	Préciser	Remarques ou informations supplémentaires
AWS CLI v2	Oui	
SDK pour C++	Oui	
SDK pour Go V2 (1.x)	Oui	
SDK pour Go 1.x (V1)	Oui	
SDK pour Java 2.x	Oui	
SDK pour Java 1.x	Partie	Amazon EKS Pod Identity et AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE non pris en charge.
SDK pour 3.x JavaScript	Oui	
SDK pour 2.x JavaScript	Oui	
SDK pour Kotlin	Oui	
SDK pour .NET 3.x	Oui	
SDK pour PHP 3.x	Oui	
SDK pour Python (Boto3)	Oui	
SDK pour Ruby 3.x	Oui	
SDK pour Rust	Oui	
Outils pour PowerShell	Oui	

Fournisseur d'identifiants IAM Identity Center

Ce mécanisme d'authentification permet AWS IAM Identity Center d'obtenir un accès par authentification unique (SSO) Services AWS à votre code.

Note

Dans la documentation de l'API du AWS SDK, le fournisseur d'informations d'identification IAM Identity Center est appelé fournisseur d'informations d'identification SSO.

Après avoir activé IAM Identity Center, vous définissez un profil pour ses paramètres dans votre AWS `config` fichier partagé. Ce profil est utilisé pour se connecter au portail d'accès IAM Identity Center. Lorsqu'un utilisateur s'authentifie avec succès auprès d'IAM Identity Center, le portail renvoie des informations d'identification à court terme pour le rôle IAM associé à cet utilisateur. Pour savoir comment le SDK obtient des informations d'identification temporaires à partir de la configuration et les utilise pour les Service AWS demandes, consultez [Comprendre l'authentification IAM Identity Center](#).

Il existe deux manières de configurer IAM Identity Center via le `config` fichier :

- Configuration du fournisseur de jetons SSO (recommandée) : durée de session prolongée.
- Configuration ancienne non actualisable : utilise une session fixe de huit heures.

Dans les deux configurations, vous devez vous reconnecter à l'expiration de votre session.

Pour définir des durées de session personnalisées, vous devez utiliser la configuration du fournisseur de jetons SSO.

Les deux guides suivants contiennent des informations supplémentaires sur IAM Identity Center :

- [AWS IAM Identity Center Guide de l'utilisateur](#)
- [AWS IAM Identity Center Référence de l'API du portail](#)

Prérequis

Vous devez d'abord activer IAM Identity Center. Pour plus de détails sur l'activation de l'authentification IAM Identity Center, consultez [Getting Started](#) dans le guide de AWS IAM Identity Center l'utilisateur.

Vous pouvez également suivre les [Authentification IAM Identity Center](#) instructions de ce guide. Ces instructions fournissent des conseils complets, allant de l'activation d'IAM Identity Center à la réalisation de la configuration des `config` fichiers partagés requise ci-dessous.

Configuration du fournisseur de jetons SSO

Note

Pour utiliser le AWS CLI afin de créer cette configuration pour vous, voir [Configurer votre profil à l'aide de l'aws configure ssoassistant](#) dans le AWS CLI.

Lorsque vous utilisez la configuration du fournisseur de jetons SSO, votre AWS SDK ou outil actualise automatiquement votre session jusqu'à votre période de session prolongée. Pour plus d'informations sur la durée de session et la durée maximale, voir [Configurer la durée de session du portail d' AWS accès et des applications intégrées d'IAM Identity Center](#) dans le guide de l'AWS IAM Identity Center utilisateur.

La `sso-session` section du config fichier est utilisée pour regrouper les variables de configuration permettant d'acquérir des jetons d'accès SSO, qui peuvent ensuite être utilisés pour acquérir des AWS informations d'identification. Pour plus de détails sur le formatage des sections d'un config fichier, consultez [Format du fichier de configuration](#).

Vous définissez une `sso-session` section et vous l'associez à un profil. `sso_region` et `sso_start_url` doit être défini dans la `sso-session` section. Généralement, `sso_account_id` et `sso_role_name` doit être défini dans la `profile` section afin que le SDK puisse demander des AWS informations d'identification.

Note

Pour en savoir plus sur la façon dont les SDK et les outils utilisent et actualisent les informations d'identification à l'aide de cette configuration, voir [Comprendre l'authentification IAM Identity Center](#).

L'exemple suivant configure le SDK pour demander les informations d'identification du IAM Identity Center. Il prend également en charge l'actualisation automatique des jetons.

```
[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
```

Vous pouvez réutiliser sso-session les configurations sur plusieurs profils.

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

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

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

sso_account_id et sso_role_name ne sont pas obligatoires pour tous les scénarios de configuration de jetons SSO. Si votre application utilise uniquement Services AWS cette authentification par support, les informations d' AWS identification traditionnelles ne sont pas nécessaires. L'authentification du porteur est un schéma d'authentification HTTP qui utilise des jetons de sécurité appelés jetons porteurs. Dans ce scénario, sso_account_id et sso_role_name ne sont pas obligatoires. Consultez le guide individuel pour savoir Service AWS s'il prend en charge l'autorisation par jeton au porteur.

Les étendues d'enregistrement sont configurées dans le cadre d'un sso-session. Scope est un mécanisme permettant OAuth 2.0 de limiter l'accès d'une application au compte d'un utilisateur. Une application peut demander une ou plusieurs étendues, et le jeton d'accès délivré à l'application est limité aux étendues accordées. Ces étendues définissent les autorisations demandées pour être autorisées pour le client OIDC enregistré et les jetons d'accès récupérés par le client. Pour connaître les options d'étendue d'accès prises en charge, consultez la section [Étendue d'accès](#) dans le guide de AWS IAM Identity Center l'utilisateur. L'exemple suivant fournit un sso_registration_scopes accès pour répertorier les comptes et les rôles.

```
[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
```

Le jeton d'authentification est mis en cache sur le disque sous le `~/.aws/sso/cache` répertoire avec un nom de fichier basé sur le nom de session.

Configuration ancienne non actualisable

L'actualisation automatique des jetons n'est pas prise en charge avec l'ancienne configuration non actualisable. Nous vous recommandons d'utiliser [Configuration du fournisseur de jetons SSO](#) plutôt le.

Pour utiliser l'ancienne configuration non actualisable, vous devez spécifier les paramètres suivants dans votre profil :

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

Vous spécifiez le portail utilisateur d'un profil avec les `sso_region` paramètres `sso_start_url` et. Vous spécifiez les autorisations à l'aide `sso_account_id` des `sso_role_name` paramètres et.

L'exemple suivant définit les quatre valeurs requises dans le config fichier.

```
[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
```

Le jeton d'authentification est mis en cache sur le disque sous le `~/.aws/sso/cache` répertoire avec un nom de fichier basé sur `lesso_start_url`.

Paramètres du fournisseur d'informations d'identification IAM Identity Center

Configurez cette fonctionnalité à l'aide des méthodes suivantes :

sso_start_url- réglage AWS **config** du fichier partagé

URL pointant vers le portail d'accès à l'IAM Identity Center de votre organisation. Pour plus d'informations sur le portail d'accès à l'IAM Identity Center, consultez la section [Utilisation du portail AWS d'accès](#) dans le guide de l'AWS IAM Identity Center utilisateur.

Pour trouver cette valeur, ouvrez la [console IAM Identity Center, consultez](#) le tableau de bord et recherchez l'URL du portail AWS d'accès.

sso_region- réglage AWS **config** du fichier partagé

Celui Région AWS qui contient l'hôte de votre portail IAM Identity Center, c'est-à-dire la région que vous avez sélectionnée avant d'activer IAM Identity Center. Elle est indépendante de votre AWS région par défaut et peut être différente.

Pour une liste complète des Régions AWS et de leurs codes, consultez la section [Points de terminaison régionaux](#) dans le Référence générale d'Amazon Web Services. Pour trouver cette valeur, ouvrez la [console IAM Identity Center, consultez](#) le tableau de bord et recherchez Region.

sso_account_id- réglage AWS **config** du fichier partagé

L'ID numérique du Compte AWS qui a été ajouté via le AWS Organizations service à utiliser pour l'authentification.

Pour consulter la liste des comptes disponibles, accédez à la [console IAM Identity Center](#) et ouvrez la Comptes AWSpage. Vous pouvez également consulter la liste des comptes disponibles à l'aide de la méthode [ListAccounts](#)API dans le Guide de référence des API du AWS IAM Identity Center portail. Par exemple, vous pouvez appeler la AWS CLI méthode [list-accounts](#).

sso_role_name- réglage AWS **config** du fichier partagé

Nom d'un ensemble d'autorisations fourni en tant que rôle IAM qui définit les autorisations obtenues par l'utilisateur. Le rôle doit exister dans le champ Compte AWS spécifié par `sso_account_id`. Utilisez le nom du rôle, et non le nom Amazon Resource Name (ARN) du rôle.

Les ensembles d'autorisations sont associés à des politiques IAM et à des politiques d'autorisations personnalisées et définissent le niveau d'accès des utilisateurs à ce qui leur est attribué Comptes AWS.

Pour voir la liste des ensembles d'autorisations disponibles par Compte AWS, accédez à la [console IAM Identity Center](#) et ouvrez la Comptes AWSpage. Choisissez le nom du jeu d'autorisations correct répertorié dans le Comptes AWS tableau. Vous pouvez

également consulter la liste des ensembles d'autorisations disponibles à l'aide de la méthode [ListAccountRoles](#) API dans le manuel de référence des API du AWS IAM Identity Center portail. Par exemple, vous pouvez appeler la AWS CLI méthode [list-account-roles](#).

sso_registration_scopes- réglage AWS **config** du fichier partagé

Une liste séparée par des virgules de chaînes de portée valides à autoriser pour. `sso-session`
Les étendues autorisent l'accès aux points de terminaison autorisés par le porteur du jeton IAM Identity Center. Une portée minimale de `sso:account:access` doit être accordée pour récupérer un jeton d'actualisation auprès du service IAM Identity Center. Pour connaître les chaînes d'étendue d'accès prises en charge, consultez la section [Étendue d'accès](#) dans le guide de AWS IAM Identity Center l'utilisateur. Ce paramètre ne s'applique pas à l'ancienne configuration non actualisable. Les jetons émis à l'aide de l'ancienne configuration sont `sso:account:access` implicitement limités à leur portée.

Compatibilité avec les AWS SDK

Les SDK suivants prennent en charge les fonctionnalités et les paramètres décrits dans cette rubrique. Toute exception partielle est notée. Tous les paramètres de propriété du système JVM sont pris en charge par le AWS SDK for Java et le Kit AWS SDK pour Kotlin seul.

Kit SDK	Présent	Remarques ou informations supplémentaires
AWS CLI v2	Oui	
SDK pour C++	Oui	
SDK pour Go V2 (1.x)	Oui	
SDK pour Go 1.x (V1)	Oui	Pour utiliser les paramètres des <code>config</code> fichiers partagés, vous devez activer le chargement à partir du fichier de configuration ; voir Sessions .
SDK pour Java 2.x	Oui	Les valeurs de configuration sont également prises en charge dans <code>credentials</code> le fichier.
SDK pour Java 1.x	Non	

Kit SDK	Pr	Remarques ou informations supplémentaires
SDK pour 3.x JavaScript	Oui	
SDK pour 2.x JavaScript	Oui	
SDK pour Kotlin	Oui	
SDK pour .NET 3.x	Oui	
SDK pour PHP 3.x	Oui	
SDK pour Python (Boto3)	Oui	
SDK pour Ruby 3.x	Oui	
SDK pour Rust	Partie	Configuration ancienne non actualisable uniquement.
Outils pour PowerShell	Oui	

fournisseur d'informations d'identification IMDS

Le service de métadonnées d'instance (IMDS) fournit des données sur votre instance que vous pouvez utiliser pour configurer ou gérer l'instance en cours d'exécution. Pour plus d'informations sur les données disponibles, consultez la section [Métadonnées d'instance et données utilisateur](#) dans le guide de l'utilisateur Amazon EC2 ou [Métadonnées d'instance et données utilisateur](#) dans le guide de l'utilisateur Amazon EC2. Amazon EC2 fournit un point de terminaison local à la disposition des instances qui peut fournir différentes informations à l'instance. Si un rôle est attaché à l'instance, elle peut fournir un ensemble d'informations d'identification valides pour ce rôle. Les SDK peuvent utiliser ce point de terminaison pour résoudre les informations d'identification dans le cadre de leur chaîne de [fournisseurs d'informations d'identification par défaut](#). La version 2 du service de métadonnées d'instance (IMDSv2), une version plus sécurisée d'IMDS qui utilise un jeton de session, est utilisée par défaut. Si cela échoue en raison d'une condition non réessayable (codes d'erreur HTTP 403, 404, 405), IMDSv1 est utilisé comme solution de secours.

Configurez cette fonctionnalité à l'aide des méthodes suivantes :

AWS_EC2_METADATA_DISABLED- variable d'environnement

Essayer ou non d'utiliser le service de métadonnées d'instance Amazon EC2 (IMDS) pour obtenir des informations d'identification.

Valeur par défaut : `false`.

Valeurs valides:

- **true**— N'utilisez pas l'IMDS pour obtenir des informations d'identification.
- **false**— Utilisez IMDS pour obtenir des informations d'identification.

ec2_metadata_v1_disabled- réglage AWS **config** du fichier partagé,

AWS_EC2_METADATA_V1_DISABLED- variable d'environnement, **aws.disableEc2MetadataV1**-

Propriété du système JVM : Java/Kotlin uniquement

S'il faut ou non utiliser le service de métadonnées d'instance version 1 (IMDSv1) comme solution de secours en cas d'échec d'IMDSv2.

Note

Les nouveaux SDK ne sont pas compatibles avec IMDSv1 et ne prennent donc pas en charge ce paramètre. Pour plus de détails, voir le tableau [Compatibilité avec les AWS SDK](#).

Valeur par défaut : `false`.

Valeurs valides:

- **true**— N'utilisez pas IMDSv1 comme solution de secours.
- **false**— Utilisez IMDSv1 comme solution de secours.

ec2_metadata_service_endpoint- réglage AWS **config** du fichier

partagé, **AWS_EC2_METADATA_SERVICE_ENDPOINT**- variable d'environnement,

aws.ec2MetadataServiceEndpoint- Propriété du système JVM : Java/Kotlin uniquement

Le point final de l'IMDS.

Valeur par défaut : si elle `ec2_metadata_service_endpoint_mode` est égale `IPv4`, le point de terminaison par défaut est `http://169.254.169.254`. Si

`ec2_metadata_service_endpoint_mode` égal IPv6, le point de terminaison par défaut est `http://[fd00:ec2::254]`.

Valeurs valides : URI valide.

`ec2_metadata_service_endpoint_mode`- réglage AWS **config** du fichier partagé, **`AWS_EC2_METADATA_SERVICE_ENDPOINT_MODE`**- variable d'environnement, **`aws.ec2MetadataServiceEndpointMode`**- Propriété du système JVM : Java/Kotlin uniquement

Le mode endpoint de l'IMDS.

Valeur par défaut : IPv4.

Valeurs valides : IPv4, IPv6.

Note

Le fournisseur d'informations d'identification IMDS fait partie du [Chaîne de fournisseurs d'identifiants](#). Cependant, le fournisseur d'informations d'identification IMDS n'est vérifié qu'après plusieurs autres fournisseurs de cette série. Par conséquent, si vous souhaitez que votre programme utilise les informations d'identification de ce fournisseur, vous devez supprimer les autres fournisseurs d'informations d'identification valides de votre configuration ou utiliser un profil différent. Sinon, au lieu de vous fier à la chaîne de fournisseurs d'informations d'identification pour découvrir automatiquement quel fournisseur renvoie des informations d'identification valides, spécifiez l'utilisation du fournisseur d'informations d'identification IMDS dans le code. Vous pouvez spécifier les sources d'informations d'identification directement lorsque vous créez des clients de service.

Sécurité des informations d'identification IMDS

Par défaut, lorsque le AWS SDK n'est pas configuré avec des informations d'identification valides, le SDK tente d'utiliser le service de métadonnées d'instance Amazon EC2 (IMDS) pour récupérer les informations d'identification d'un rôle. AWS Ce comportement peut être désactivé en définissant la variable d'environnement `AWS_EC2_METADATA_DISABLED` sur `true`. Cela empêche toute activité réseau inutile et améliore la sécurité sur les réseaux non fiables sur lesquels le service de métadonnées d'instance Amazon EC2 peut être usurpé.

Note

AWS Les clients du SDK configurés avec des informations d'identification valides n'utiliseront jamais IMDS pour récupérer des informations d'identification, quels que soient ces paramètres.

Désactivation de l'utilisation des informations d'identification Amazon EC2 IMDS

La façon dont vous définissez cette variable d'environnement dépend du système d'exploitation utilisé et du fait que vous souhaitez ou non que la modification soit persistante.

Linux et macOS

Les clients utilisant Linux ou macOS peuvent définir cette variable d'environnement à l'aide de la commande suivante :

```
$ export AWS_EC2_METADATA_DISABLED=true
```

Si vous souhaitez que ce paramètre soit permanent pendant plusieurs sessions et redémarrages du système, vous pouvez ajouter la commande ci-dessus à votre fichier de profil shell, telle que `.bash_profile`, `.zsh_profile`, ou `.profile`.

Windows

Les clients utilisant Windows peuvent définir cette variable d'environnement à l'aide de la commande suivante :

```
$ set AWS_EC2_METADATA_DISABLED=true
```

Si vous souhaitez que ce paramètre soit persistant sur plusieurs sessions shell et redémarrages du système, vous pouvez utiliser la commande suivante à la place :

```
$ setx AWS_EC2_METADATA_DISABLED=true
```

Note

La `setx` commande n'applique pas la valeur à la session shell en cours. Vous devrez donc recharger ou rouvrir le shell pour que la modification soit prise en compte.

Compatibilité avec les AWS SDK

Les SDK suivants prennent en charge les fonctionnalités et les paramètres décrits dans cette rubrique. Toute exception partielle est notée. Tous les paramètres de propriété du système JVM sont pris en charge par le AWS SDK for Java et le Kit AWS SDK pour Kotlin seul.

SDK	Prise en charge	Remarques ou informations supplémentaires
AWS CLI v2	Oui	
SDK pour C++	Oui	
SDK pour Go V2 (1.x)	Oui	
SDK pour Go 1.x (V1)	Oui	Pour utiliser les paramètres des config fichiers partagés, vous devez activer le chargement à partir du fichier de configuration ; voir Sessions .
SDK pour Java 2.x	Oui	
SDK pour Java 1.x	Partie	Propriétés du système JVM : à utiliser à la place de <code>aws.sdk.disableEc2MetadataV1</code> ; <code>aws.ec2MetadataServiceEndpoint</code> et <code>aws.ec2MetadataServiceEndpointMode</code> non prises en charge.
SDK pour 3.x JavaScript	Oui	
SDK pour 2.x JavaScript	Oui	
SDK pour Kotlin	Oui	N'utilise pas la solution de secours IMDSv1.
SDK pour .NET 3.x	Oui	
SDK pour PHP 3.x	Oui	
SDK pour Python (Boto3)	Oui	

SDK	Préférences	Remarques ou informations supplémentaires
SDK pour Ruby 3.x	Oui	
SDK pour Rust	Oui	N'utilise pas la solution de secours IMDSv1.
Outils pour PowerShell	Oui	Vous pouvez désactiver le repli IMDSv1 de manière explicite dans le code en utilisant. <code>[Amazon.Util.EC2InstanceMetadata]::EC2MetadataV1Disabled = \$true</code>

Fournisseur d'identifiants de processus

Les SDK permettent d'étendre la chaîne de fournisseurs d'informations d'identification pour des cas d'utilisation personnalisés.

IAM Roles Anywhere permet d'obtenir des informations d'identification temporaires pour une charge de travail ou un processus qui s'exécute en dehors de AWS. `credential_process` Pour configurer cet usage, voir [Rôles Anywhere IAM](#).

Warning

Ce qui suit décrit une méthode d'obtention d'informations d'identification auprès d'un processus externe. Cela peut être dangereux, alors soyez prudent. Dans la mesure du possible, il convient de privilégier d'autres fournisseurs d'informations d'identification. Si vous utilisez cette option, vous devez vous assurer que le config fichier est aussi verrouillé que possible conformément aux meilleures pratiques de sécurité de votre système d'exploitation. Vérifiez que votre outil d'identification personnalisé n'écrit aucune information secrète `stderr`, car les SDK AWS CLI peuvent capturer et enregistrer ces informations, les exposant potentiellement à des utilisateurs non autorisés.

Configurez cette fonctionnalité à l'aide des méthodes suivantes :

`credential_process`- réglage AWS `config` du fichier partagé

Spécifie une commande externe que le SDK ou l'outil exécute en votre nom pour générer ou récupérer les informations d'authentification à utiliser. Le paramètre spécifie le nom d'un programme/commande que le SDK appellera. Lorsque le SDK appelle le processus, il attend que celui-ci y écrive des données JSON. `stdout` Le fournisseur personnalisé doit renvoyer les informations dans un format spécifique. Ces informations contiennent les informations d'identification que le SDK ou l'outil peut utiliser pour vous authentifier.

Note

Le fournisseur d'informations d'identification du processus fait partie du [Chaîne de fournisseurs d'identifiants](#). Cependant, le fournisseur d'identifiants de processus n'est vérifié qu'après plusieurs autres fournisseurs de cette série. Par conséquent, si vous souhaitez que votre programme utilise les informations d'identification de ce fournisseur, vous devez supprimer les autres fournisseurs d'informations d'identification valides de votre configuration ou utiliser un profil différent. Sinon, au lieu de vous fier à la chaîne de fournisseurs d'informations d'identification pour découvrir automatiquement quel fournisseur renvoie des informations d'identification valides, spécifiez l'utilisation du fournisseur d'informations d'identification de processus dans le code. Vous pouvez spécifier les sources d'informations d'identification directement lorsque vous créez des clients de service.

Spécification du chemin d'accès au programme d'identification

La valeur du paramètre est une chaîne qui contient le chemin d'accès à un programme que le SDK ou l'outil de développement exécute en votre nom :

- Le chemin et le nom du fichier ne peuvent être composés que des caractères suivants : A-Z, a-z, 0-9, tiret (-), trait de soulignement (_), point (.), barre oblique (/), barre oblique inverse (\) et espace.
- Si le chemin d'accès ou le nom du fichier contient un espace, entourez le chemin d'accès complet et le nom du fichier de guillemets doubles (" ").
- Si un nom de paramètre ou une valeur de paramètre contient un espace, entourez cet élément de guillemets doubles (" "). Entourez uniquement le nom ou la valeur, pas la paire.
- N'incluez aucune variable d'environnement dans les chaînes. Par exemple, n'incluez pas `$HOME` ou `%USERPROFILE%`.

- Ne spécifiez pas le dossier de base comme `~`. * Vous devez spécifier le chemin complet ou le nom du fichier de base. S'il existe un nom de fichier de base, le système tente de trouver le programme dans les dossiers spécifiés par la variable d'PATHEnvironnement.

L'exemple suivant montre comment définir `credential_process` dans le config fichier partagé sous Linux/macOS.

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

L'exemple suivant montre comment définir `credential_process` dans le config fichier partagé sous Windows.

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

Sortie valide du programme d'identification

Le SDK exécute la commande comme indiqué dans le profil, puis lit les données du flux de sortie standard. La commande que vous spécifiez, qu'il s'agisse d'un script ou d'un programme binaire, doit générer une sortie JSON STDOUT correspondant à la syntaxe suivante.

```
{  
  "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

A la date de cette publication, la clé `Version` doit être définie sur 1. Cela pourrait augmenter avec le temps à mesure que la structure évolue.

La `Expiration` clé est un horodatage au format RFC3339. Si la `Expiration` clé n'est pas présente dans la sortie de l'outil, le SDK suppose que les informations d'identification sont des

informations d'identification à long terme qui ne sont pas actualisées. Dans le cas contraire, les informations d'identification sont considérées comme des informations d'identification temporaires et elles sont automatiquement actualisées en réexécutant la `credential_process` commande avant leur expiration.

Note

Le SDK ne met pas en cache les informations d'identification des processus externes de la même manière qu'il assume les informations d'identification des rôles. Si la mise en cache est obligatoire, vous devez la mettre en œuvre dans le processus externe.

Le processus externe peut renvoyer un code de retour non nul pour indiquer qu'une erreur s'est produite lors de la récupération des informations d'identification.

Compatibilité avec les AWS SDK

Les SDK suivants prennent en charge les fonctionnalités et les paramètres décrits dans cette rubrique. Toute exception partielle est notée. Tous les paramètres de propriété du système JVM sont pris en charge par le AWS SDK for Java et le Kit AWS SDK pour Kotlin seul.

Kit SDK	Préciser	Remarques ou informations supplémentaires
AWS CLI v2	Oui	
SDK pour C++	Oui	
SDK pour Go V2 (1.x)	Oui	
SDK pour Go 1.x (V1)	Oui	Pour utiliser les paramètres des config fichiers partagés, vous devez activer le chargement à partir du fichier de configuration ; voir Sessions .
SDK pour Java 2.x	Oui	
SDK pour Java 1.x	Oui	

Kit SDK	Pré-requis	Remarques ou informations supplémentaires
SDK pour 3.x JavaScript	Oui	
SDK pour 2.x JavaScript	Oui	
SDK pour Kotlin	Oui	
SDK pour .NET 3.x	Oui	
SDK pour PHP 3.x	Oui	
SDK pour Python (Boto3)	Oui	
SDK pour Ruby 3.x	Oui	
SDK pour Rust	Oui	
Outils pour PowerShell	Oui	

AWS Fonctionnalités standardisées des SDK et des outils

De nombreuses fonctionnalités ont été standardisées selon des valeurs par défaut cohérentes et pour fonctionner de la même manière dans de nombreux SDK. Cette cohérence augmente la productivité et la clarté lors du codage sur plusieurs SDK. Tous les paramètres peuvent être remplacés dans le code. Consultez l'API de votre SDK spécifique pour plus de détails.

Important

Les SDK ne prennent pas tous en charge toutes les fonctionnalités, ni même tous les aspects d'une fonctionnalité.

Rubriques

- [ID d'application](#)
- [Métadonnées d'instance Amazon EC2.](#)

- [Points d'accès Amazon S3](#)
- [Amazon S3 Multi-Region Access Points](#)
- [Région AWS](#)
- [AWS STS Points de terminaison régionalisés](#)
- [Points de terminaison à double pile et FIPS](#)
- [Découverte du points de terminaison](#)
- [Paramètres de configuration généraux](#)
- [Client IMDS](#)
- [Comportement de nouvelle tentative](#)
- [Compression des demandes](#)
- [Points de terminaison spécifiques au service](#)
- [Paramètres de configuration intelligents par défaut](#)

ID d'application

Un seul Compte AWS peut être utilisé par plusieurs applications clients pour passer des appels à Services AWS. L'ID d'application permet aux clients d'identifier quelle application source a effectué un ensemble d'appels à l'aide d'un Compte AWS. AWS Les SDK et les services n'utilisent ni n'interprètent cette valeur autrement que pour la réintégrer dans les communications avec les clients. Par exemple, cette valeur peut être incluse dans les e-mails opérationnels ou dans le AWS Health Dashboard pour identifier de manière unique laquelle de vos applications est associée à la notification.

Configurez cette fonctionnalité à l'aide des méthodes suivantes :

sdk_ua_app_id- réglage AWS **config** du fichier partagé, **AWS_SDK_UA_APP_ID**- variable d'environnement, **aws.userAgentAppId**- Propriété du système JVM : Java/Kotlin uniquement

Ce paramètre est une chaîne unique que vous assignez à votre application pour identifier les applications auxquelles une Compte AWS application donnée appelle AWS.

Valeur par défaut : None

Valeurs valides : chaîne d'une longueur maximale de 50. Les lettres, les chiffres et les caractères spéciaux suivants sont autorisés : ! \$ %& * + , - . / : ; ^ _ ` | ~.

Exemple de définition de cette valeur dans le config fichier :

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

Exemple Linux/macOS de définition de variables d'environnement via la ligne de commande :

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

Exemple Windows de définition de variables d'environnement via la ligne de commande :

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

Si vous incluez des symboles qui ont une signification particulière pour le shell utilisé, évitez la valeur le cas échéant.

Compatibilité avec les AWS SDK

Les SDK suivants prennent en charge les fonctionnalités et les paramètres décrits dans cette rubrique. Toute exception partielle est notée. Tous les paramètres de propriété du système JVM sont pris en charge par le AWS SDK for Java et le Kit AWS SDK pour Kotlin seul.

SDK	Pré-requis	Remarques ou informations supplémentaires
AWS CLI v2	Nor	
SDK pour C++	Oui	config file partagé n'est pas pris en charge.
SDK pour Go V2 (1.x)	Oui	
SDK pour Go 1.x (V1)	Nor	
SDK pour Java 2.x	Partiel	Le paramètre de fichier partagé n'est pas pris en charge ; la variable d'environnement n'est pas prise en charge.

SDK	Pré-requis	Remarques ou informations supplémentaires
SDK pour Java 1.x	Nor	
SDK pour 3.x JavaScript	Oui	
SDK pour 2.x JavaScript	Nor	
SDK pour Kotlin	Oui	
SDK pour .NET 3.x	Oui	Les variables d'environnement ne sont pas prises en charge.
SDK pour PHP 3.x	Oui	
SDK pour Python (Boto3)	Oui	
SDK pour Ruby 3.x	Oui	
SDK pour Rust	Oui	
Outils pour PowerShell	Nor	

Métadonnées d'instance Amazon EC2.

Amazon EC2 fournit un service sur les instances appelé Instance Metadata Service (IMDS). Pour en savoir plus sur ce service, consultez la section [Métadonnées d'instance et données utilisateur](#) dans le guide de l'utilisateur Amazon EC2 ou [Métadonnées d'instance et données utilisateur](#) dans le guide de l'utilisateur Amazon EC2. Lorsque vous tentez de récupérer des informations d'identification sur une instance Amazon EC2 configurée avec un rôle IAM, la connexion au service de métadonnées de l'instance est ajustable.

Configurez cette fonctionnalité à l'aide des méthodes suivantes :

metadata_service_num_attempts- réglage AWS **config** du fichier partagé,
AWS_METADATA_SERVICE_NUM_ATTEMPTS- variable d'environnement

Ce paramètre indique le nombre total de tentatives à effectuer avant d'abandonner lors de la tentative de récupération de données à partir du service de métadonnées d'instance.

Valeur par défaut : 1

Valeurs valides : nombre supérieur ou égal à 1.

metadata_service_timeout- réglage AWS **config** du fichier partagé,
AWS_METADATA_SERVICE_TIMEOUT- variable d'environnement

Spécifie le nombre de secondes avant l'expiration du délai lorsque vous tentez de récupérer des données à partir du service de métadonnées d'instance.

Valeur par défaut : 1

Valeurs valides : nombre supérieur ou égal à 1.

Exemple de définition de ces valeurs dans le config fichier :

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

Exemple Linux/macOS de définition de variables d'environnement via la ligne de commande :

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

Exemple Windows de définition de variables d'environnement via la ligne de commande :

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

Compatibilité avec les AWS SDK

Les SDK suivants prennent en charge les fonctionnalités et les paramètres décrits dans cette rubrique. Toute exception partielle est notée. Tous les paramètres de propriété du système JVM sont pris en charge par le AWS SDK for Java et le Kit AWS SDK pour Kotlin seul.

SDK	Pr	Remarques ou informations supplémentaires
AWS CLI v2	Oui	
SDK pour C++	Nor	
SDK pour Go V2 (1.x)	Nor	
SDK pour Go 1.x (V1)	Nor	
SDK pour Java 2.x	Nor	
SDK pour Java 1.x	Partie	Seule la clause <code>AWS_METADATA_SERVICE_TIMEOUT</code> est prise en charge.
SDK pour 3.x JavaScript	Nor	
SDK pour 2.x JavaScript	Nor	
SDK pour Kotlin	Nor	
SDK pour .NET 3.x	Nor	
SDK pour PHP 3.x	Oui	
SDK pour Python (Boto3)	Oui	
SDK pour Ruby 3.x	Nor	
SDK pour Rust	Nor	
Outils pour PowerShell	Nor	

Points d'accès Amazon S3

Le service Amazon S3 fournit des points d'accès comme autre moyen d'interagir avec les compartiments Amazon S3. Les points d'accès ont des politiques et des configurations uniques qui peuvent leur être appliquées plutôt que directement au bucket. Avec AWS les SDK, vous pouvez

utiliser les Amazon Resource Names (ARN) du point d'accès dans le champ du bucket pour les opérations d'API au lieu de spécifier explicitement le nom du bucket. Ils sont utilisés pour des opérations spécifiques telles que l'utilisation d'un point d'accès ARN [GetObject](#) pour récupérer un objet dans un bucket, ou l'utilisation d'un ARN de point d'accès avec [PutObject](#) pour ajouter un objet à un bucket.

Pour en savoir plus sur les points d'accès et les ARN Amazon S3, consultez la section [Utilisation des points d'accès](#) dans le guide de l'utilisateur Amazon S3.

Configurez cette fonctionnalité à l'aide des méthodes suivantes :

s3_use_arn_region- réglage AWS **config** du fichier partagé, **AWS_S3_USE_ARN_REGION**-variable d'environnement, **aws.s3UseArnRegion**- Propriété du système JVM : Java/Kotlin uniquement, Pour configurer la valeur directement dans le code, consultez directement votre SDK spécifique.

Ce paramètre contrôle si le SDK utilise l'ARN du point d'accès Région AWS pour créer le point de terminaison régional pour la demande. Le SDK confirme que l'ARN Région AWS est servi par la même AWS partition que celle configurée par le client afin Région AWS d'empêcher les appels entre partitions susceptibles d'échouer. Si le paramètre multiplicateur est défini, le paramètre configuré par le code est prioritaire, suivi du paramètre de variable d'environnement.

Valeur par défaut : `false`

Valeurs valides:

- **true**— Le SDK utilise les ARN Région AWS lors de la construction du point de terminaison plutôt que celui configuré Région AWS par le client. Exception : si la configuration du client Région AWS est une norme FIPS Région AWS, elle doit correspondre à l' Région AWS ARN. Si vous ne le faites pas, une erreur se produit.
- **false**— Le SDK utilise la configuration du client Région AWS lors de la construction du point de terminaison.

Compatibilité avec les AWS SDK

Les SDK suivants prennent en charge les fonctionnalités et les paramètres décrits dans cette rubrique. Toute exception partielle est notée. Tous les paramètres de propriété du système JVM sont pris en charge par le AWS SDK for Java et le Kit AWS SDK pour Kotlin seul.

SDK	Pi er ch	Remarques ou informations supplémentaires
AWS CLI v2	Oui	
SDK pour C++	Oui	
SDK pour Go V2 (1.x)	Oui	
SDK pour Go 1.x (V1)	Oui	Pour utiliser les paramètres des config fichiers partagés, vous devez activer le chargement à partir du fichier de configuration ; voir Sessions .
SDK pour Java 2.x	Oui	
SDK pour Java 1.x	Oui	La propriété du système JVM n'est pas prise en charge.
SDK pour 3.x JavaScript	Oui	
SDK pour 2.x JavaScript	Oui	
SDK pour Kotlin	Oui	
SDK pour .NET 3.x	Oui	Ne suit pas la priorité standard ; la valeur config du fichier partagé a priorité sur la variable d'environnement.
SDK pour PHP 3.x	Oui	
SDK pour Python (Boto3)	Oui	
SDK pour Ruby 3.x	Oui	
SDK pour Rust	Nor	
Outils pour PowerShell	Oui	Ne suit pas la priorité standard ; la valeur config du fichier partagé a priorité sur la variable d'environnement.

Amazon S3 Multi-Region Access Points

Les points d'accès multirégionaux Amazon S3 fournissent un point de terminaison global que les applications peuvent utiliser pour traiter les demandes provenant de compartiments Amazon S3 situés dans plusieurs compartiments. Régions AWS Vous pouvez utiliser des points d'accès multirégionaux pour créer des applications multirégionales avec la même architecture que celle utilisée dans une seule région, puis exécuter ces applications n'importe où dans le monde.

Pour en savoir plus sur les points d'accès multirégionaux, consultez la section [Points d'accès multirégionaux dans Amazon S3](#) dans le guide de l'utilisateur d'Amazon S3.

Pour en savoir plus sur les noms de ressources Amazon (ARN) des points d'accès multirégionaux, consultez la section [Faire des demandes à l'aide d'un point d'accès multirégional](#) dans le guide de l'utilisateur Amazon S3.

Pour en savoir plus sur la création de points d'accès multirégionaux, consultez la section [Gestion des points d'accès multirégionaux](#) dans le guide de l'utilisateur Amazon S3.

L'algorithme SigV4A est l'implémentation de signature utilisée pour signer les demandes régionales globales. Cet algorithme est obtenu par le SDK via une dépendance à l'égard du [AWSbibliothèques CRT \(Common Runtime\)](#).

Configurez cette fonctionnalité à l'aide des méthodes suivantes :

s3_disable_multiregion_access_points- réglage AWS **config** du fichier partagé, **AWS_S3_DISABLE_MULTIREGION_ACCESS_POINTS**- variable d'environnement, **aws.s3DisableMultiRegionAccessPoints**- Propriété du système JVM : Java/Kotlin uniquement, Pour configurer la valeur directement dans le code, consultez directement votre SDK spécifique.

Ce paramètre détermine si le SDK est susceptible de tenter des requêtes interrégionales. Si le paramètre multiplicateur est défini, le paramètre configuré par le code est prioritaire, suivi du paramètre de variable d'environnement.

Valeur par défaut : `false`

Valeurs valides:

- **true**— Arrête l'utilisation des demandes interrégionales.
- **false**— Active les demandes interrégionales à l'aide de points d'accès multirégionaux.

Compatibilité avec les AWS SDK

Les SDK suivants prennent en charge les fonctionnalités et les paramètres décrits dans cette rubrique. Toute exception partielle est notée. Tous les paramètres de propriété du système JVM sont pris en charge par le AWS SDK for Java et le Kit AWS SDK pour Kotlin seul.

Kit SDK	Préciser	Remarques ou informations supplémentaires
AWS CLI v2	Oui	
SDK pour C++	Oui	
SDK pour Go V2 (1.x)	Oui	
SDK pour Go 1.x (V1)	Non	
SDK pour Java 2.x	Oui	
SDK pour Java 1.x	Non	
SDK pour 3.x JavaScript	Oui	
SDK pour 2.x JavaScript	Non	
SDK pour Kotlin	Oui	
SDK pour .NET 3.x	Oui	
SDK pour PHP 3.x	Oui	
SDK pour Python (Boto3)	Oui	
SDK pour Ruby 3.x	Oui	
SDK pour Rust	Oui	
Outils pour PowerShell	Oui	

Région AWS

Régions AWS sont un concept important à comprendre lorsque vous travaillez avec Services AWS.

Avec Régions AWS, vous pouvez accéder à Services AWS ceux qui résident physiquement dans une zone géographique spécifique. Cela peut être utile pour que vos données et applications fonctionnent à proximité de l'endroit où vous et vos utilisateurs y accédez. Les régions fournissent une tolérance aux pannes, une stabilité et une résilience, et peuvent également réduire la latence. Avec les régions, vous pouvez créer des ressources redondantes qui restent disponibles et qui ne sont pas affectées par une panne régionale.

La plupart des Service AWS demandes sont associées à une région géographique particulière. Les ressources que vous créez dans une région n'existent dans aucune autre région, sauf si vous utilisez explicitement une fonctionnalité de réplication proposée par un Service AWS. Par exemple, Amazon S3 et Amazon EC2 prennent en charge la réplication entre régions. Certains services, tels que IAM, ne disposent pas de ressources régionales.

Références générales AWSII contient des informations sur les points suivants :

- Pour comprendre la relation entre les régions et les points de terminaison, et pour consulter la liste des points de terminaison régionaux existants, voir Points de terminaison de [AWS service](#).
- Pour consulter la liste actuelle de toutes les régions et points de terminaison pris en charge pour chacune d'entre elles Service AWS, voir Points de [terminaison et quotas de service](#).

Création de clients de service

Pour y accéder par programmation Services AWS, les SDK utilisent une classe/un objet client pour chacun. Service AWS Si votre application doit accéder à Amazon EC2, par exemple, elle créera un objet client Amazon EC2 pour établir une interface avec ce service.

Si aucune région n'est explicitement spécifiée pour le client, le client utilise par défaut la région définie par le biais du `region` paramètre suivant. Cependant, la région active d'un client peut être définie explicitement pour n'importe quel objet client individuel. La définition de la région de cette manière a priorité sur tout paramètre global pour ce client de service en particulier. La région alternative est spécifiée lors de l'instanciation de ce client, en fonction de votre SDK (consultez votre guide SDK spécifique ou la base de code de votre SDK).

Configurez cette fonctionnalité à l'aide des méthodes suivantes :

region- réglage AWS **config** du fichier partagé, **AWS_REGION**- variable d'environnement, **aws.region**- Propriété du système JVM : Java/Kotlin uniquement

Spécifie la valeur par défaut Région AWS à utiliser pour les AWS demandes. Cette région est utilisée pour les demandes de service du SDK qui ne sont pas fournies avec une région spécifique à utiliser.

Valeur par défaut : Aucune. Vous devez spécifier cette valeur de manière explicite.

Valeurs valides:

- N'importe quel code de région disponible pour le service choisi, tel qu'indiqué dans les [points de terminaison du AWS service](#) dans la référence AWS générale. Par exemple, la valeur `us-east-1` définit le point de terminaison sur l'est des Région AWS États-Unis (Virginie du Nord).
- `aws-global` spécifie le point de terminaison global pour les services qui prennent en charge un point de terminaison mondial distinct en plus des points de terminaison régionaux, tels que AWS Security Token Service (AWS STS) et Amazon Simple Storage Service (Amazon S3).

Exemple de définition de cette valeur dans le config fichier :

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

Exemple Linux/macOS de définition de variables d'environnement via la ligne de commande :

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

Exemple Windows de définition de variables d'environnement via la ligne de commande :

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

La plupart des SDK disposent d'un objet de « configuration » qui permet de définir la région par défaut à partir du code de l'application. Pour plus de détails, consultez le guide du développeur de votre AWS SDK spécifique.

Compatibilité avec les AWS SDK

Les SDK suivants prennent en charge les fonctionnalités et les paramètres décrits dans cette rubrique. Toute exception partielle est notée. Tous les paramètres de propriété du système JVM sont pris en charge par le AWS SDK for Java et le Kit AWS SDK pour Kotlin seul.

Kit SDK	Préférences	Remarques ou informations supplémentaires
AWS CLI v2	Oui	AWS CLI v2 utilise n'importe quelle valeur in <code>AWS_REGION</code> avant toute valeur in <code>AWS_DEFAULT_REGION</code> (les deux variables sont cochées).
AWS CLI v1	Oui	AWS CLI v1 utilise une variable d'environnement nommée <code>AWS_DEFAULT_REGION</code> à cette fin.
SDK pour C++	Oui	
SDK pour Go V2 (1.x)	Oui	
SDK pour Go 1.x (V1)	Oui	Pour utiliser les paramètres des config fichiers partagés, vous devez activer le chargement à partir du fichier de configuration ; voir Sessions .
SDK pour Java 2.x	Oui	
SDK pour Java 1.x	Oui	
SDK pour 3.x JavaScript	Oui	
SDK pour 2.x JavaScript	Oui	
SDK pour Kotlin	Oui	
SDK pour .NET 3.x	Oui	
SDK pour PHP 3.x	Oui	
SDK pour Python (Boto3)	Oui	Ce SDK utilise une variable d'environnement nommée <code>AWS_DEFAULT_REGION</code> à cette fin.
SDK pour Ruby 3.x	Oui	
SDK pour Rust	Oui	

Kit SDK	Préremarques ou informations supplémentaires
Outils pour PowerShell	Oui

AWS STS Points de terminaison régionalisés

Par défaut, AWS Security Token Service (AWS STS) est disponible en tant que service global, et toutes les AWS STS demandes sont envoyées à un seul point de terminaison à `https://sts.amazonaws.com`. Les demandes globales concernent la région de l'est des États-Unis (Virginie du Nord). AWS recommande d'utiliser des AWS STS points de terminaison régionaux plutôt que des points de terminaison globaux. Pour plus d'informations sur les points de terminaison AWS STS, reportez-vous à la section [Points de terminaison](#) de la référence AWS Security Token Service d'API.

Configurez cette fonctionnalité à l'aide des méthodes suivantes :

sts_regional_endpoints- réglage AWS **config** du fichier partagé,

AWS_STS_REGIONAL_ENDPOINTS- variable d'environnement

Ce paramètre indique comment le SDK ou l'outil détermine le Service AWS point de terminaison qu'il utilise pour communiquer avec le AWS Security Token Service (AWS STS).

Valeur par défaut : `legacy`

Note

Toutes les nouvelles versions majeures du SDK publiées après juillet 2022 seront par défaut définies `surregional`. Les nouvelles versions majeures du SDK peuvent supprimer ce paramètre et utiliser `regional` le comportement. Pour réduire l'impact futur de cette modification, nous vous recommandons de commencer à l'utiliser `regional` dans votre application dès que possible.

Valeurs valides : (Valeur recommandée : `regional`)

- **legacy**— Utilise le point de terminaison global `sts.amazonaws.com` pour les régions suivantes : `ap-south-1`, `ap-southeast-1`, `ap-southeast-2`, `aws-global`, `ca-central-1`, `eu-central-1`, `eu-north-1`, `eu-west-1`, `eu-west-2`, `eu-west-3`, `sa-east-1`, `us-east-1`, `us-east-2`, `us-west-1`, `us-west-2`

west-2,eu-west-3,sa-east-1,us-east-1,us-east-2,us-west-1, etus-west-2. Toutes les autres régions utilisent automatiquement leur point de terminaison régional respectif.

- **regional**— Le SDK ou l'outil utilise toujours le AWS STS point de terminaison de la région actuellement configurée. Par exemple, si le client est configuré pour être utilisé sur us-west-2, tous les appels AWS STS sont effectués vers le point de terminaison régional sts.us-west-2.amazonaws.com, plutôt que vers le point de terminaison global sts.amazonaws.com. Pour envoyer une demande au point de terminaison global lorsque ce paramètre est activé, vous pouvez définir la région sur aws-global.

Exemple de définition de ces valeurs dans le config fichier :

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

Exemple Linux/macOS de définition de variables d'environnement via la ligne de commande :

```
export AWS_STS_REGIONAL_ENDPOINTS=regional
```

Exemple Windows de définition de variables d'environnement via la ligne de commande :

```
setx AWS_STS_REGIONAL_ENDPOINTS regional
```

Compatibilité avec les AWS SDK

Les SDK suivants prennent en charge les fonctionnalités et les paramètres décrits dans cette rubrique. Toute exception partielle est notée. Tous les paramètres de propriété du système JVM sont pris en charge par le AWS SDK for Java et le Kit AWS SDK pour Kotlin seul.

Kit SDK	Partie	Remarques ou informations supplémentaires
AWS CLI v2	Partie	La valeur par défaut est <code>regional</code> .
SDK pour C++	Partie	Les variables d'environnement et les paramètres de config fichier ne sont pas pris en charge. Le SDK fonctionne avec les <code>regional</code> paramètres.

Kit SDK	Pré-requis	Remarques ou informations supplémentaires
SDK pour Go V2 (1.x)	Oui	
SDK pour Go 1.x (V1)	Oui	Pour utiliser les paramètres des config fichiers partagés, vous devez activer le chargement à partir du fichier de configuration ; voir Sessions .
SDK pour Java 2.x	Oui	
SDK pour Java 1.x	Oui	
SDK pour 3.x JavaScript	Oui	
SDK pour 2.x JavaScript	Oui	
SDK pour Kotlin	Non	
SDK pour .NET 3.x	Oui	
SDK pour PHP 3.x	Oui	
SDK pour Python (Boto3)	Oui	
SDK pour Ruby 3.x	Oui	
SDK pour Rust	Oui	
Outils pour PowerShell	Oui	

Points de terminaison à double pile et FIPS

Configurez cette fonctionnalité à l'aide des méthodes suivantes :

use_dualstack_endpoint- réglage AWS **config** du fichier partagé, **AWS_USE_DUALSTACK_ENDPOINT**- variable d'environnement, **aws.useDualstackEndpoint**- Propriété du système JVM : Java/Kotlin uniquement

Active ou désactive l'envoi de demandes par le SDK aux points de terminaison à double pile. Pour en savoir plus sur les points de terminaison à double pile, qui prennent en charge le trafic IPv4 et IPv6, consultez la section Utilisation des [points de terminaison à double pile Amazon S3](#) dans le guide de l'utilisateur d'Amazon Simple Storage Service. Des points de terminaison à double pile sont disponibles pour certains services dans certaines régions.

Valeur par défaut : `false`

Valeurs valides:

- **true**— Le SDK ou l'outil tentera d'utiliser des points de terminaison à double pile pour effectuer des requêtes réseau. S'il n'existe pas de point de terminaison à double pile pour le service et/ou Région AWS la demande échouera.
- **false**— Le SDK ou l'outil n'utilisera pas de points de terminaison à double pile pour effectuer des requêtes réseau.

use_fips_endpoint- réglage AWS **config** du fichier partagé, **AWS_USE_FIPS_ENDPOINT**- variable d'environnement, **aws.useFipsEndpoint**- Propriété du système JVM : Java/Kotlin uniquement

Active ou désactive l'envoi de demandes par le SDK ou l'outil à des points de terminaison conformes à la norme FIPS. Les normes fédérales de traitement de l'information (FIPS) sont un ensemble d'exigences de sécurité du gouvernement américain relatives aux données et à leur cryptage. Les agences gouvernementales, les partenaires et ceux qui souhaitent faire affaire avec le gouvernement fédéral sont tenus de respecter les directives FIPS. Contrairement aux AWS terminaux standard, les points de terminaison FIPS utilisent une bibliothèque logicielle TLS conforme à la norme FIPS 140-2. Si ce paramètre est activé et qu'il n'existe pas de point de terminaison FIPS pour le service dans votre ordinateur Région AWS, l' AWS appel peut échouer. [Points de terminaison spécifiques au service](#) et l' `--endpoint-url` option permettant de AWS Command Line Interface remplacer ce paramètre.

Pour en savoir plus sur les autres méthodes de spécification des points de terminaison FIPS par Région AWS, consultez la section Points de [terminaison FIPS](#) par service. Pour plus d'informations sur les points de terminaison du service Amazon Elastic Compute Cloud, consultez la section Points de terminaison [à double pile \(IPv4 et IPv6\) dans le manuel](#) Amazon EC2 API Reference.

Valeur par défaut : `false`

Valeurs valides:

- **true**— Le SDK ou l'outil enverra des demandes aux terminaux conformes à la norme FIPS.
- **false**— Le SDK ou l'outil n'enverra pas de demandes aux terminaux conformes à la norme FIPS.

Compatibilité avec les AWS SDK

Les SDK suivants prennent en charge les fonctionnalités et les paramètres décrits dans cette rubrique. Toute exception partielle est notée. Tous les paramètres de propriété du système JVM sont pris en charge par le AWS SDK for Java et le Kit AWS SDK pour Kotlin seul.

Kit SDK	Prise en charge	Remarques ou informations supplémentaires
AWS CLI v2	Oui	
SDK pour C++	Oui	
SDK pour Go V2 (1.x)	Oui	
SDK pour Go 1.x (V1)	Oui	Pour utiliser les paramètres des config fichiers partagés, vous devez activer le chargement à partir du fichier de configuration ; voir Sessions .
SDK pour Java 2.x	Oui	
SDK pour Java 1.x	Non	
SDK pour 3.x JavaScript	Oui	
SDK pour 2.x JavaScript	Oui	
SDK pour Kotlin	Oui	
SDK pour .NET 3.x	Oui	

Kit SDK	Pré-requis	Remarques ou informations supplémentaires
SDK pour PHP 3.x	Oui	
SDK pour Python (Boto3)	Oui	
SDK pour Ruby 3.x	Oui	
SDK pour Rust	Oui	
Outils pour PowerShell	Oui	

Découverte des points de terminaison

Les SDK utilisent la découverte des points de terminaison pour accéder aux points de terminaison des services (URL permettant d'accéder à diverses ressources), tout en conservant la flexibilité nécessaire pour modifier les URL AWS selon les besoins. De cette façon, votre code peut détecter automatiquement les nouveaux points de terminaison. Il n'existe aucun point de terminaison fixe pour certains services. Au lieu de cela, vous obtenez les points de terminaison disponibles pendant l'exécution en faisant d'abord une demande pour obtenir les points de terminaison. Après avoir récupéré les points de terminaison disponibles, le code utilise le point de terminaison pour accéder à d'autres opérations. Par exemple, pour Amazon Timestream, le SDK fait `DescribeEndpoints` une demande pour récupérer les points de terminaison disponibles, puis utilise ces points de terminaison pour effectuer des opérations spécifiques telles que `CreateDatabase` ou `CreateTable`.

Configurez cette fonctionnalité à l'aide des méthodes suivantes :

`endpoint_discovery_enabled`- réglage AWS `config` du fichier partagé, **`AWS_ENABLE_ENDPOINT_DISCOVERY`**- variable d'environnement, **`aws.endpointDiscoveryEnabled`**- Propriété du système JVM : Java/Kotlin uniquement, Pour configurer la valeur directement dans le code, consultez directement votre SDK spécifique.

Active ou désactive la découverte des points de terminaison pour DynamoDB.

La découverte des terminaux est obligatoire dans Timestream et facultative dans Amazon DynamoDB. Ce paramètre est défini par défaut sur l'un `true` ou l'autre `false` selon que le

service nécessite ou non la découverte des points de terminaison. Les requêtes Timestream sont par défaut et les `true` requêtes Amazon DynamoDB sont définies par défaut sur `false`

Valeurs valides:

- **true**— Le SDK doit automatiquement tenter de découvrir un point de terminaison pour les services où la découverte du point de terminaison est facultative.
- **false**— Le SDK ne doit pas tenter automatiquement de découvrir un point de terminaison pour les services où la découverte de point de terminaison est facultative.

Compatibilité avec les AWS SDK

Les SDK suivants prennent en charge les fonctionnalités et les paramètres décrits dans cette rubrique. Toute exception partielle est notée. Tous les paramètres de propriété du système JVM sont pris en charge par le AWS SDK for Java et le Kit AWS SDK pour Kotlin seul.

SDK	Pr er ch	Remarques ou informations supplémentaires
AWS CLI v2	Oui	
SDK pour C++	Oui	
SDK pour Go V2 (1.x)	Oui	
SDK pour Go 1.x (V1)	Oui	Pour utiliser les paramètres des config fichiers partagés, vous devez activer le chargement à partir du fichier de configuration ; voir Sessions .
SDK pour Java 2.x	Oui	Le SDK pour Java 2.x <code>AWS_ENDPOINT_DISCOVERY_ENABLED</code> utilise comme nom de variable d'environnement.
SDK pour Java 1.x	Partie	La propriété du système JVM n'est pas prise en charge.
SDK pour 3.x JavaScript	Oui	
SDK pour 2.x JavaScript	Oui	
SDK pour Kotlin	Oui	

SDK	Pr	Remarques ou informations supplémentaires
SDK pour .NET 3.x	Oui	
SDK pour PHP 3.x	Oui	
SDK pour Python (Boto3)	Oui	
SDK pour Ruby 3.x	Oui	
SDK pour Rust	Partie	Pris en charge uniquement pour Timestream.
Outils pour PowerShell	Oui	

Paramètres de configuration généraux

Les SDK prennent en charge certains paramètres généraux qui configurent les comportements généraux des SDK.

Configurez cette fonctionnalité à l'aide des méthodes suivantes :

api_versions- réglage AWS **config** du fichier partagé

Certains AWS services gèrent plusieurs versions d'API afin de garantir la rétrocompatibilité. Par défaut, le SDK et les AWS CLI opérations utilisent la dernière version d'API disponible. Pour exiger l'utilisation d'une version d'API spécifique pour vos demandes, incluez le `api_versions` paramètre dans votre profil.

Valeur par défaut : Aucune. (La dernière version de l'API est utilisée par le SDK.)

Valeurs valides : il s'agit d'un paramètre imbriqué suivi d'une ou de plusieurs lignes en retrait identifiant chacune un AWS service et la version d'API à utiliser. Consultez la documentation du AWS service pour savoir quelles versions d'API sont disponibles.

L'exemple définit une version d'API spécifique pour deux AWS services du `config` fichier. Ces versions de l'API ne sont utilisées que pour les commandes qui s'exécutent sous le profil qui contient ces paramètres. Les commandes de tout autre service utilisent la dernière version de l'API de ce service.


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

ca_bundle- réglage AWS **config** du fichier partagé, **AWS_CA_BUNDLE**- variable d'environnement

Spécifie le chemin d'accès à un ensemble de certificats personnalisé (un fichier avec une .pem extension) à utiliser lors de l'établissement de connexions SSL/TLS.

Valeur par défaut : aucune

Valeurs valides : Spécifiez le chemin complet ou le nom du fichier de base. S'il existe un nom de fichier de base, le système tente de trouver le programme dans les dossiers spécifiés par la variable d'PATHenvironnement.

Exemple de définition de cette valeur dans le config fichier :

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

Exemple Linux/macOS de définition de variables d'environnement via la ligne de commande :

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

Exemple Windows de définition de variables d'environnement via la ligne de commande :

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

output- réglage AWS **config** du fichier partagé

Spécifie la manière dont les résultats sont formatés dans le AWS SDK AWS CLI et dans les autres outils.

Valeur par défaut : json

Valeurs valides:

- **json**— La sortie est formatée sous forme de chaîne [JSON](#).
- **yaml**— La sortie est formatée sous forme de chaîne [YAML](#).

- **[yaml-stream](#)**— La sortie est diffusée et formatée sous forme de chaîne [YAML](#). Le streaming permet de traiter plus rapidement de gros types de données.
- **[text](#)**— La sortie est formatée sous la forme de plusieurs lignes de valeurs de chaîne séparées par des tabulations. Cela peut être utile pour transmettre le résultat à un processeur de texte, comme `grep`, `sed`, ou `awk`.
- **[table](#)**— La sortie est formatée sous forme de tableau en utilisant les caractères `+|-` pour former les bordures des cellules. La présentation des informations est dans un format beaucoup plus lisible par l'utilisateur que les autres, mais peu pratique du point de vue programmation.

parameter_validation- réglage AWS **config** du fichier partagé

Spécifie si le SDK ou l'outil tente de valider les paramètres de ligne de commande avant de les envoyer au point de terminaison du AWS service.

Valeur par défaut : `true`

Valeurs valides:

- **`true`** – Valeur par défaut Le SDK ou l'outil effectue la validation côté client des paramètres de ligne de commande. Cela permet au SDK ou à l'outil de confirmer que les paramètres sont valides et de détecter certaines erreurs. Le SDK ou l'outil peut rejeter les demandes non valides avant de les envoyer au point de terminaison du AWS service.
- **`false`**— Le SDK ou l'outil ne valide pas les paramètres de ligne de commande avant de les envoyer au point de terminaison du AWS service. Le point AWS de terminaison du service est chargé de valider toutes les demandes et de rejeter les demandes non valides.

Compatibilité avec les AWS SDK

Les SDK suivants prennent en charge les fonctionnalités et les paramètres décrits dans cette rubrique. Toute exception partielle est notée. Tous les paramètres de propriété du système JVM sont pris en charge par le AWS SDK for Java et le Kit AWS SDK pour Kotlin seul.

Kit SDK	Remarques ou informations supplémentaires
AWS CLI v2	Partie <code>api_versions</code> non pris en charge.
SDK pour C++	Oui

Kit SDK	Pr	Remarques ou informations supplémentaires
SDK pour Go V2 (1.x)	Partie	<code>api_versions</code> et <code>parameter_validation</code> non pris en charge.
SDK pour Go 1.x (V1)	Partie	<code>api_versions</code> et <code>parameter_validation</code> non pris en charge. Pour utiliser les paramètres des config fichiers partagés, vous devez activer le chargement à partir du fichier de configuration ; voir Sessions .
SDK pour Java 2.x	Nor	
SDK pour Java 1.x	Nor	
SDK pour 3.x JavaScript	Oui	
SDK pour 2.x JavaScript	Oui	
SDK pour Kotlin	Nor	
SDK pour .NET 3.x	Nor	
SDK pour PHP 3.x	Oui	
SDK pour Python (Boto3)	Oui	
SDK pour Ruby 3.x	Oui	
SDK pour Rust	Nor	
Outils pour PowerShell	Nor	

Client IMDS

Les SDK implémentent un client IMDSv2 (Instance Metadata Service Version 2) à l'aide de requêtes orientées session. Pour plus d'informations sur IMDSv2, consultez [Utiliser IMDSv2](#) dans le guide de l'utilisateur Amazon EC2 ou Utiliser [IMDSv2 dans](#) le guide de l'utilisateur Amazon EC2. Le client IMDS est configurable via un objet de configuration client disponible dans la base de code du SDK.

Configurez cette fonctionnalité à l'aide des méthodes suivantes :

retries- membre de l'objet de configuration client

Le nombre de nouvelles tentatives pour chaque demande ayant échoué.

Valeur par défaut : 3

Valeurs valides : nombre supérieur à 0.

port- membre de l'objet de configuration client

Port du point de terminaison.

Valeur par défaut : 80

Valeurs valides : Nombre.

token_ttl- membre de l'objet de configuration client

Le TTL du jeton.

Valeur par défaut : 21 600 secondes (6 heures, durée maximale allouée).

Valeurs valides : Nombre.

endpoint- membre de l'objet de configuration client

Le point final de l'IMDS.

Valeur par défaut : si elle `endpoint_mode` est égale `IPv4`, le point de terminaison par défaut est `http://169.254.169.254`. Si `endpoint_mode` égal `IPv6`, le point de terminaison par défaut est `http://[fd00:ec2::254]`.

Valeurs valides : URI valide.

Les options suivantes sont prises en charge par la plupart des SDK. Consultez la base de code de votre SDK spécifique pour plus de détails.

endpoint_mode- membre de l'objet de configuration client

Le mode endpoint de l'IMDS.

Valeur par défaut : IPv4

Valeurs valides : IPv4, IPv6

http_open_timeout- membre de l'objet de configuration client (le nom peut varier)

Le nombre de secondes à attendre avant l'ouverture de la connexion.

Valeur par défaut : 1 seconde.

Valeurs valides : nombre supérieur à 0.

http_read_timeout- membre de l'objet de configuration client (le nom peut varier)

Le nombre de secondes nécessaires à la lecture d'un bloc de données.

Valeur par défaut : 1 seconde.

Valeurs valides : nombre supérieur à 0.

http_debug_output- membre de l'objet de configuration client (le nom peut varier)

Définit un flux de sortie pour le débogage.

Valeur par défaut : Aucune.

Valeurs valides : un flux d'E/S valide, tel que STDOUT.

backoff- membre de l'objet de configuration client (le nom peut varier)

Le nombre de secondes passées en veille entre deux tentatives ou le nombre de secondes qu'un client a fourni une fonction de temporisation pour appeler. Cela remplace la stratégie de ralentissement exponentiel par défaut.

Valeur par défaut : varie selon le SDK.

Valeurs valides : varie selon le SDK. Il peut s'agir d'une valeur numérique ou d'un appel à une fonction personnalisée.

Compatibilité avec les AWS SDK

Les SDK suivants prennent en charge les fonctionnalités et les paramètres décrits dans cette rubrique. Toute exception partielle est notée. Tous les paramètres de propriété du système JVM sont pris en charge par le AWS SDK for Java et le Kit AWS SDK pour Kotlin seul.

SDK	Préférences	Remarques ou informations supplémentaires
AWS CLI v2	Oui	
SDK pour C++	Non	IMDSv2 utilisé en interne uniquement. veuillez consulter fournisseur d'informations d'identification IMDS .
SDK pour Go V2 (1.x)	Oui	
SDK pour Go 1.x (V1)	Oui	
SDK pour Java 2.x	Oui	
SDK pour Java 1.x	Oui	
SDK pour 3.x JavaScript	Oui	
SDK pour 2.x JavaScript	Oui	
SDK pour Kotlin	Oui	
SDK pour .NET 3.x	Oui	
SDK pour PHP 3.x	Oui	
SDK pour Python (Boto3)	Oui	
SDK pour Ruby 3.x	Oui	
SDK pour Rust	Oui	
Outils pour PowerShell	Oui	

Comportement de nouvelle tentative

Le comportement des nouvelles tentatives inclut les paramètres relatifs à la manière dont les SDK tentent de se rétablir après des échecs résultant de demandes adressées à Services AWS

Configurez cette fonctionnalité à l'aide des méthodes suivantes :

max_attempts- réglage AWS **config** du fichier partagé, **AWS_MAX_ATTEMPTS**- variable d'environnement, **aws.maxAttempts**- Propriété du système JVM : Java/Kotlin uniquement

Spécifie le nombre maximal de tentatives à effectuer sur une demande.

Valeur par défaut : si cette valeur n'est pas spécifiée, sa valeur par défaut dépend de la valeur du `retry_mode` paramètre :

- Si `retry_mode` est le cas `legacy` — Utilise une valeur par défaut spécifique à votre SDK (consultez le guide de votre SDK spécifique ou la base de code de votre SDK pour `max_attempts` connaître les valeurs par défaut).
- Si `retry_mode` c'est le cas `standard` — Fait trois tentatives.
- Si `retry_mode` c'est le cas `adaptive` — Fait trois tentatives.

Valeurs valides : nombre supérieur à 0.

retry_mode- réglage AWS **config** du fichier partagé, **AWS_RETRY_MODE**- variable d'environnement, **aws.retryMode**- Propriété du système JVM : Java/Kotlin uniquement

Spécifie la manière dont le SDK ou l'outil de développement tente de réessayer.

Valeur par défaut : `legacy` est la stratégie de nouvelle tentative par défaut.

Valeurs valides:

- `legacy`— Spécifique à votre SDK (consultez le guide de votre SDK spécifique ou la base de code de votre SDK).
- `standard`— L'ensemble standard de règles de nouvelle tentative pour tous les AWS SDK. Ce mode inclut un ensemble standard d'erreurs réessayées et prend en charge les quotas de nouvelles tentatives. Le nombre maximum de tentatives par défaut avec ce mode est de trois, sauf si cela `max_attempts` est explicitement configuré.
- `adaptive`— Un mode de nouvelle tentative expérimental qui inclut les fonctionnalités du mode `standard` mais inclut une régulation automatique côté client. Ce mode étant expérimental, il est susceptible de modifier le comportement à l'avenir.

Choix entre les **standard** modes et **adaptive** nouvelle tentative

Nous vous recommandons d'utiliser le mode `standard` nouvelle tentative, sauf si vous êtes certain que votre utilisation vous convient le mieux. `adaptive`

Note

Le `adaptive` mode suppose que vous regroupez des clients en fonction de l'étendue dans laquelle le service principal peut limiter les demandes. Si vous ne le faites pas, les limitations d'une ressource peuvent retarder les demandes pour une ressource non liée si vous utilisez le même client pour les deux ressources.

Standard	Adaptatif
Cas d'utilisation des applications : Tous.	Cas d'utilisation des applications : <ol style="list-style-type: none"> 1. Insensible à la latence. 2. Le client n'accède qu'à une seule ressource , ou vous fournissez une logique pour regrouper vos clients séparément en fonction de la ressource de service à laquelle il accède.
Supporte le coupe-circuit pour empêcher le SDK de réessayer en cas de panne.	Supporte le coupe-circuit pour empêcher le SDK de réessayer en cas de panne.
Utilise un recul exponentiel agité en cas de panne.	Utilise des durées d'interruption dynamiques pour tenter de minimiser le nombre de demandes ayant échoué, en échange du risque d'augmentation de la latence.
Ne retarde jamais la première tentative de demande, uniquement les nouvelles tentatives.	Peut limiter ou retarder la tentative de demande initiale.

Si vous choisissez d'utiliser `adaptive` le mode, votre application doit créer des clients conçus en fonction de chaque ressource susceptible d'être limitée. Dans ce cas, une ressource est plus fine que de simplement penser à chacune d'elles. Service AWS Services AWS peuvent avoir des dimensions supplémentaires qu'ils utilisent pour limiter les demandes. Prenons l'exemple du service Amazon DynamoDB. DynamoDB Région AWS utilise plus la table accessible pour limiter les demandes. Cela signifie qu'une table à laquelle votre code accède peut être plus limitée que d'autres. Si votre code utilise le même client pour accéder à toutes les tables et que les demandes adressées à l'une de

ces tables sont limitées, le mode de nouvelle tentative adaptatif réduira le taux de demandes pour toutes les tables. Votre code doit être conçu pour avoir un client par région-et-table paire R. Si vous rencontrez une latence inattendue lors de l'utilisation du adaptative mode, consultez le guide de AWS documentation spécifique au service que vous utilisez.

Détails de mise en œuvre du mode Réessai

Voici le pseudocode de haut niveau pour les modes standard et adaptative nouvelle tentative :

```
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)
  }
}
```

Vous trouverez ci-dessous de plus amples informations sur les composants utilisés dans le pseudocode :

GetSendToken:

Les compartiments à jetons ne sont utilisés qu'en mode adaptative nouvelle tentative. Les buckets de jetons imposent un taux de demandes maximal en exigeant qu'un jeton soit disponible pour lancer une demande. Le client SDK est configurable pour échouer rapidement à la demande ou le bloquer jusqu'à ce qu'un jeton soit disponible.

La limitation du débit côté client est un algorithme qui permet initialement d'effectuer des demandes à n'importe quel taux, jusqu'à concurrence de l'allocation de jetons. Cependant, une fois qu'une réponse limitée est détectée, le client rate-of-request est alors limité en conséquence. L'allocation de jetons est également augmentée en conséquence si des réponses positives sont reçues.

Grâce à la limitation de débit adaptative, les SDK peuvent ralentir le débit d'envoi des demandes afin de mieux répondre à la capacité de Services AWS.

SendHTTPRequest:

La plupart AWS des SDK utilisent une bibliothèque HTTP qui utilise des pools de connexions afin que vous puissiez réutiliser une connexion existante lorsque vous effectuez une requête HTTP. En général, les connexions sont réutilisées lors d'une nouvelle tentative de demande en raison d'erreurs de régulation. Les demandes ne sont pas réutilisées lors d'une nouvelle tentative en raison d'erreurs transitoires.

RequestBookkeeping:

Le quota de nouvelles tentatives doit être mis à jour si la demande aboutit. Pour le mode `adaptive` nouvelle tentative uniquement, la variable d'état `maxsendrate` est mise à jour en fonction du type de réponse reçue.

Retryable:

Cette étape détermine si une réponse peut être réessayée en fonction des éléments suivants :

- Le code d'état HTTP .
- Le code d'erreur renvoyé par le service.
- Erreurs de connexion, définies comme toute erreur reçue par le SDK sans qu'une réponse HTTP du service n'ait été reçue.

Les erreurs transitoires (codes d'état HTTP 400, 408, 500, 502, 503 et 504) et les erreurs de régulation (codes d'état HTTP 400, 403, 429, 502, 503 et 509) peuvent toutes être réessayées. Le comportement des nouvelles tentatives du SDK est déterminé en combinaison avec les codes d'erreur ou d'autres données du service.

MAX_ATTEMPTS:

Spécifié par le paramètre `config` du fichier ou la variable d'environnement.

HasRetryQuota

Cette étape limite les demandes de nouvelle tentative en exigeant qu'un jeton soit disponible dans le compartiment de quotas de nouvelles tentatives. Les compartiments de quotas de nouvelles tentatives sont un mécanisme destiné à empêcher les nouvelles tentatives qui ont peu de chances de réussir. Ces quotas dépendent du SDK, dépendent souvent du client et parfois même des points

de terminaison du service. Les jetons de quota de nouvelles tentatives disponibles sont supprimés lorsque les demandes échouent pour diverses raisons, et réapprovisionnés lorsqu'elles aboutissent. Lorsqu'il ne reste aucun jeton, la boucle de nouvelles tentatives est interrompue.

ExponentialBackoff

Dans le cas d'une erreur pouvant être tentée à nouveau, le délai de nouvelle tentative est calculé à l'aide d'une réduction exponentielle tronquée. Les SDK utilisent un ralentissement exponentiel binaire tronqué avec gigue. L'algorithme suivant montre comment la durée de sommeil, en secondes, est définie pour une réponse à une demande i :

$$\text{seconds_to_sleep_i} = \min(b * r^i, \text{MAX_BACKOFF})$$

Dans l'algorithme précédent, les valeurs suivantes s'appliquent :

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

r = 2

MAX_BACKOFF = 20 secondspour la plupart des SDK. Consultez le guide ou le code source de votre SDK spécifique pour obtenir une confirmation.

Compatibilité avec les AWS SDK

Les SDK suivants prennent en charge les fonctionnalités et les paramètres décrits dans cette rubrique. Toute exception partielle est notée. Tous les paramètres de propriété du système JVM sont pris en charge par le AWS SDK for Java et le Kit AWS SDK pour Kotlin seul.

SDK	Préremarques ou informations supplémentaires
AWS CLI v2	Oui
SDK pour C++	Oui
SDK pour Go V2 (1.x)	Oui
SDK pour Go 1.x (V1)	Nor
SDK pour Java 2.x	Oui

SDK	Pr	Remarques ou informations supplémentaires
SDK pour Java 1.x	Oui	Propriétés du système JVM : utiliser <code>com.amazonaws.sdk.maxAttempts</code> au lieu de <code>aws.maxAttempts</code> ; utiliser <code>com.amazonaws.sdk.retryMode</code> au lieu de <code>aws.retryMode</code> .
SDK pour 3.x JavaScript	Oui	
SDK pour 2.x JavaScript	Non	Prend en charge un nombre maximal de tentatives, un recul exponentiel avec instabilité et une option pour une méthode personnalisée d'annulation des nouvelles tentatives.
SDK pour Kotlin	Oui	
SDK pour .NET 3.x	Oui	
SDK pour PHP 3.x	Oui	
SDK pour Python (Boto3)	Oui	
SDK pour Ruby 3.x	Oui	
SDK pour Rust	Oui	
Outils pour PowerShell	Oui	

Compression des demandes

AWS Les SDK et les outils peuvent automatiquement compresser les charges utiles lors de l'envoi de demandes au support recevant Services AWS des charges utiles compressées. La compression de la charge utile du client avant de l'envoyer à un service peut réduire le nombre total de demandes et la bande passante nécessaires pour envoyer des données au service, ainsi que le nombre de demandes infructueuses en raison des limites du service relatives à la taille de la charge utile. Pour la compression, le SDK ou l'outil sélectionne un algorithme de codage pris en charge à la fois par le service et par le SDK. Cependant, la liste actuelle des encodages possibles se compose uniquement de gzip, mais elle pourrait s'étendre à l'avenir.

La compression des demandes peut être particulièrement utile si votre application utilise [Amazon CloudWatch](#). CloudWatch est un service de surveillance et d'observabilité qui collecte des données opérationnelles et de surveillance sous forme de journaux, de métriques et d'événements. La méthode [PutMetricDataAPI](#) est un exemple d'opération de service qui prend en charge CloudWatch la compression.

Configurez cette fonctionnalité à l'aide des méthodes suivantes :

disable_request_compression- réglage AWS **config** du fichier partagé, **AWS_DISABLE_REQUEST_COMPRESSION**- variable d'environnement, **aws.disableRequestCompression**- Propriété du système JVM : Java/Kotlin uniquement

Active ou désactive la compression d'une charge utile par le SDK ou l'outil avant d'envoyer une demande.

Valeur par défaut : `false`

Valeurs valides:

- **true**— Désactive la compression des demandes.
- **false**— Utilisez la compression des demandes lorsque cela est possible.

request_min_compression_size_bytes- réglage AWS **config** du fichier partagé, **AWS_REQUEST_MIN_COMPRESSION_SIZE_BYTES**- variable d'environnement, **aws.requestMinCompressionSizeBytes**- Propriété du système JVM : Java/Kotlin uniquement

Définit la taille minimale en octets du corps de la demande que le SDK ou l'outil doit compresser. Les petites charges utiles peuvent devenir plus longues lorsqu'elles sont compressées. Il existe donc une limite inférieure à laquelle il est judicieux de procéder à la compression. Cette valeur est inclusive, une taille de demande supérieure ou égale à la valeur est compressée.

Valeur par défaut : 10240 octets

Valeurs valides : valeur entière comprise entre 0 et 10485760 octets inclus.

Compatibilité avec les AWS SDK

Les SDK suivants prennent en charge les fonctionnalités et les paramètres décrits dans cette rubrique. Toute exception partielle est notée. Tous les paramètres de propriété du système JVM sont pris en charge par le AWS SDK for Java et le Kit AWS SDK pour Kotlin seul.

SDK	Préciser	Remarques ou informations supplémentaires
AWS CLI v2	Oui	
SDK pour C++	Oui	
SDK pour Go V2 (1.x)	Oui	
SDK pour Go 1.x (V1)	Non	
SDK pour Java 2.x	Oui	
SDK pour Java 1.x	Non	
SDK pour 3.x JavaScript	Oui	
SDK pour 2.x JavaScript	Non	
SDK pour Kotlin	Oui	
SDK pour .NET 3.x	Oui	
SDK pour PHP 3.x	Oui	
SDK pour Python (Boto3)	Oui	
SDK pour Ruby 3.x	Oui	
SDK pour Rust	Oui	
Outils pour PowerShell	Oui	

Points de terminaison spécifiques au service

La configuration du point de terminaison spécifique au service offre la possibilité d'utiliser un point de terminaison de votre choix pour les demandes d'API et de conserver ce choix. Ces paramètres offrent la flexibilité nécessaire pour prendre en charge les points de terminaison locaux, les points de terminaison VPC et les environnements de développement locaux tiers AWS . Différents points

de terminaison peuvent être utilisés pour les environnements de test et de production. Vous pouvez spécifier une URL de point de terminaison pour un individu Services AWS.

Configurez cette fonctionnalité à l'aide des méthodes suivantes :

endpoint_url- réglage AWS **config** du fichier partagé, **AWS_ENDPOINT_URL**- variable d'environnement, **aws.endpointUrl**- Propriété du système JVM : Java/Kotlin uniquement

Lorsqu'il est spécifié directement dans un profil ou en tant que variable d'environnement, ce paramètre indique le point de terminaison utilisé pour toutes les demandes de service. Ce point de terminaison est remplacé par tout point de terminaison spécifique au service configuré.

Vous pouvez également utiliser ce paramètre dans une `services` section d'un AWS `config` fichier partagé pour définir un point de terminaison personnalisé pour un service spécifique. Pour obtenir la liste de toutes les clés d'identification de service à utiliser pour les sous-sections de `services` cette section, consultez [Identifiants pour les points de terminaison spécifiques au service](#).

Valeur par défaut : none

Valeurs valides : URL incluant le schéma et l'hôte du point de terminaison. L'URL peut éventuellement contenir un composant de chemin contenant un ou plusieurs segments de chemin.

AWS_ENDPOINT_URL_<SERVICE>- variable d'environnement,
aws.endpointUrl<ServiceName>- Propriété du système JVM : Java/Kotlin uniquement

AWS_ENDPOINT_URL_<SERVICE>, où **<SERVICE>** est l' Service AWS identifiant, définit un point de terminaison personnalisé pour un service spécifique. Pour obtenir la liste de toutes les variables d'environnement spécifiques au service, consultez. [Identifiants pour les points de terminaison spécifiques au service](#)

Ce point de terminaison spécifique au service remplace tout point de terminaison global défini dans. **AWS_ENDPOINT_URL**

Valeur par défaut : none

Valeurs valides : URL incluant le schéma et l'hôte du point de terminaison. L'URL peut éventuellement contenir un composant de chemin contenant un ou plusieurs segments de chemin.

ignore_configured_endpoint_urls- réglage AWS **config** du fichier partagé, **AWS_IGNORE_CONFIGURED_ENDPOINT_URLS**- variable d'environnement, **aws.ignoreConfiguredEndpointUrls**- Propriété du système JVM : Java/Kotlin uniquement

Ce paramètre est utilisé pour ignorer toutes les configurations de points de terminaison personnalisées.

Notez que tout point de terminaison explicite défini dans le code ou sur un client de service lui-même est utilisé quel que soit ce paramètre. Par exemple, l'inclusion du paramètre de ligne de `--endpoint-url` commande dans une AWS CLI commande ou la transmission d'une URL de point de terminaison à un constructeur client prendra toujours effet.

Valeur par défaut : `false`

Valeurs valides:

- **true**— Le SDK ou l'outil ne lit aucune option de configuration personnalisée à partir du `config` fichier partagé ou des variables d'environnement pour définir l'URL d'un point de terminaison.
- **false**— Le SDK ou l'outil utilise tous les points de terminaison disponibles fournis par l'utilisateur à partir du `config` fichier partagé ou des variables d'environnement.

Configuration des points de terminaison à l'aide de variables d'environnement

Pour acheminer les demandes de tous les services vers une URL de point de terminaison personnalisée, définissez la variable d'environnement `AWS_ENDPOINT_URL` globale.

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

Pour acheminer les demandes d'une URL de point de terminaison spécifique Service AWS vers une URL de point de terminaison personnalisée, utilisez la variable d'`AWS_ENDPOINT_URL_<SERVICE>` environnement. Amazon DynamoDB a un `serviceId` de [DynamoDB](#). Pour ce service, la variable d'environnement de l'URL du point de terminaison est `AWS_ENDPOINT_URL_DYNAMODB`. Ce point de terminaison a priorité sur le point de terminaison global défini `AWS_ENDPOINT_URL` pour ce service.

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


Comme autre exemple, AWS Elastic Beanstalk possède un `serviceId` de [Elastic Beanstalk](#). L' Service AWS identifiant est basé sur le modèle d'API `serviceId` en remplaçant tous les espaces par des traits de soulignement et en majuscules toutes les lettres. Pour définir le point de terminaison de ce service, la variable d'environnement correspondante est `AWS_ENDPOINT_URL_ELASTIC_BEANSTALK`. Pour obtenir la liste de toutes les variables d'environnement spécifiques au service, consultez [Identifiants pour les points de terminaison spécifiques au service](#)

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

Configuration des points de terminaison à l'aide du fichier partagé **config**

Dans le `config` fichier partagé, `endpoint_url` il est utilisé à différents endroits pour différentes fonctionnalités.

- `endpoint_url` spécifié directement dans `a profile` fait de ce point de terminaison le point de terminaison global.
- `endpoint_url` imbriqué sous une clé d'identification de service dans une `services` section, ce point de terminaison s'applique aux demandes adressées uniquement à ce service. Pour plus de détails sur la définition d'une `services` section dans votre `config` fichier partagé, consultez [Format du fichier de configuration](#).

L'exemple suivant utilise une `services` définition pour configurer une URL de point de terminaison spécifique au service à utiliser pour Amazon S3 et un point de terminaison global personnalisé à utiliser pour tous les autres services :

```
[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 profil unique peut configurer des points de terminaison pour plusieurs services. Cet exemple montre comment définir les URL de point de terminaison spécifiques au service pour Amazon S3 et AWS Elastic Beanstalk dans le même profil. AWS Elastic Beanstalk a un `serviceId` de [Elastic Beanstalk](#). L' Service AWS identifiant est basé sur le modèle d'API `serviceId` en

remplaçant tous les espaces par des traits de soulignement et en minuscules toutes les lettres. Ainsi, la clé d'identification du service devient `elastic_beanstalk` et les paramètres de ce service commencent sur la ligne `elastic_beanstalk =`. Pour obtenir la liste de toutes les clés d'identification de service à utiliser dans `services` cette section, consultez [Identifiants pour les points de terminaison spécifiques au service](#).

```
[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 section de configuration du service peut être utilisée à partir de plusieurs profils. Par exemple, deux profils peuvent utiliser la même `services` définition tout en modifiant d'autres propriétés de profil :

```
[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
```

Configurer les points de terminaison dans les profils à l'aide d'informations d'identification basées sur les rôles

Si votre profil possède des informations d'identification basées sur les rôles configurées via un `source_profile` paramètre pour la fonctionnalité IAM d'assumer un rôle, le SDK utilise uniquement les configurations de service pour le profil spécifié. Il n'utilise pas de profils auxquels des rôles sont liés. Par exemple, en utilisant le `config` fichier partagé suivant :

```
[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
```

Si vous utilisez le profil B et que vous appelez votre code à Amazon EC2, le point de terminaison est résolu comme `https://profile-b-ec2-endpoint.aws`. Si votre code envoie une demande à un autre service, la résolution du point de terminaison ne suivra aucune logique personnalisée. Le point de terminaison ne correspond pas au point de terminaison global défini dans le profil A. Pour qu'un point de terminaison global prenne effet pour B le profil, vous devez le définir `endpoint_url` directement dans le profil B. Pour plus d'informations sur le paramètre `source_profile`, consultez [Assumer le rôle de fournisseur d'informations d'identification](#).

Priorité des paramètres

Les paramètres de cette fonctionnalité peuvent être utilisés simultanément, mais une seule valeur sera prioritaire par service. Pour les appels d'API effectués vers une valeur donnée Service AWS, l'ordre suivant est utilisé pour sélectionner une valeur :

1. Tout paramètre explicite défini dans le code ou sur un client de service lui-même a priorité sur tout autre paramètre.
 - Pour le AWS CLI, il s'agit de la valeur fournie par le paramètre de ligne de `--endpoint-url` commande. Pour un SDK, les attributions explicites peuvent prendre la forme d'un paramètre que vous définissez lorsque vous instanciez un Service AWS client ou un objet de configuration.
2. La valeur fournie par une variable d'environnement spécifique au service, telle que `AWS_ENDPOINT_URL_DYNAMODB`.
3. La valeur fournie par la variable d'environnement `AWS_ENDPOINT_URL` globale du point de terminaison.
4. Valeur fournie par le `endpoint_url` paramètre imbriqué sous une clé d'identification de service dans une `services` section du config fichier partagé.

5. La valeur fournie par le `endpoint_url` paramètre spécifié directement dans un `profile` config fichier partagé.
6. Toute URL de point de terminaison par défaut pour le point correspondant Service AWS est utilisée en dernier.

Compatibilité avec les AWS SDK

Les SDK suivants prennent en charge les fonctionnalités et les paramètres décrits dans cette rubrique. Toute exception partielle est notée. Tous les paramètres de propriété du système JVM sont pris en charge par le AWS SDK for Java et le Kit AWS SDK pour Kotlin seul.

SDK	Pr er ct	Remarques ou informations supplémentaires
AWS CLI v2	Oui	
SDK pour C++	Nor	
SDK pour Go V2 (1.x)	Oui	
SDK pour Go 1.x (V1)	Nor	
SDK pour Java 2.x	Nor	
SDK pour Java 1.x	Nor	
SDK pour 3.x JavaScript	Oui	
SDK pour 2.x JavaScript	Nor	
SDK pour Kotlin	Oui	
SDK pour .NET 3.x	Oui	
SDK pour PHP 3.x	Oui	
SDK pour Python (Boto3)	Oui	
SDK pour Ruby 3.x	Oui	

SDK	Pré	Remarques ou informations supplémentaires
SDK pour Rust	Non	
Outils pour PowerShell	Oui	

Identifiants pour les points de terminaison spécifiques au service

Pour plus d'informations sur comment et où utiliser les identifiants du tableau suivant, reportez-vous à [Points de terminaison spécifiques au service](#).

serviceId	Cl	variable d'environnement AWS_ENDPOINT_URL_< SERVICE>
AccessAnalyzer	ac	AWS_ENDPOINT_URL_ACCESSANALYZER
Account	ac	AWS_ENDPOINT_URL_ACCOUNT
ACM	ac	AWS_ENDPOINT_URL_ACM
ACM PCA	ac	AWS_ENDPOINT_URL_ACM_PCA
Alexa For Business	af	AWS_ENDPOINT_URL_ALEXA_FOR_BUSINESS

serviceId	Cl	variable d'environnement AWS_ENDPOINT_URL_< d' SERVICE>
amp	ar	AWS_ENDPOINT_URL_AMP
Amplify	ar	AWS_ENDPOINT_URL_AMPLIFY
AmplifyBackend	ar	AWS_ENDPOINT_URL_AMPLIFYBACKEND
AmplifyUIBuilder	ar	AWS_ENDPOINT_URL_AMPLIFYUIBUILDER
API Gateway	ap	AWS_ENDPOINT_URL_API_GATEWAY
ApiGatewayManageme ntApi	ap	AWS_ENDPOINT_URL_APIGATEWAYMANAGEMENTAPI
ApiGatewayV2	ap	AWS_ENDPOINT_URL_APIGATEWAYV2
AppConfig	ap	AWS_ENDPOINT_URL_APPCONFIG
AppConfigData	ap	AWS_ENDPOINT_URL_APPCONFIGDATA

serviceId	Clé de la variable d'environnement AWS_ENDPOINT_URL_< SERVICE>
AppFabric	aws_endpoint_url_appfabric
Appflow	aws_endpoint_url_appflow
AppIntegrations	aws_endpoint_url_appintegrations
Application Auto Scaling	aws_endpoint_url_application_auto_scaling
Application Insights	aws_endpoint_url_application_insights
ApplicationCostProfiler	aws_endpoint_url_applicationcostprofiler
App Mesh	aws_endpoint_url_app_mesh
AppRunner	aws_endpoint_url_apprunner

serviceId	Cl	variable d'environnement AWS_ENDPOINT_URL_< d' SERVICE>
AppStream	a)	AWS_ENDPOINT_URL_APPSTREAM
AppSync	a)	AWS_ENDPOINT_URL_APPSsync
ARC Zonal Shift	a:	AWS_ENDPOINT_URL_ARC_ZONAL_SHIFT _:
Artifact	a:	AWS_ENDPOINT_URL_ARTIFACT
Athena	a:	AWS_ENDPOINT_URL_ATHENA
AuditManager	a:	AWS_ENDPOINT_URL_AUDITMANAGER g:
Auto Scaling	a:	AWS_ENDPOINT_URL_AUTO_SCALING i:
Auto Scaling Plans	a:	AWS_ENDPOINT_URL_AUTO_SCALING_PLANS i:
b2bi	b:	AWS_ENDPOINT_URL_B2BI
Backup	b:	AWS_ENDPOINT_URL_BACKUP

serviceId	Cl variable d'environnement AWS_ENDPOINT_URL_< d' SERVICE> ic: de se pc Al co le fic pa
Backup Gateway	b: AWS_ENDPOINT_URL_BACKUP_GATEWAY t:
BackupStorage	b: AWS_ENDPOINT_URL_BACKUPSTORAGE r:
Batch	b: AWS_ENDPOINT_URL_BATCH
BCM Data Exports	b: AWS_ENDPOINT_URL_BCM_DATA_EXPORTS e:
Bedrock	b: AWS_ENDPOINT_URL_BEDROCK
Bedrock Agent	b: AWS_ENDPOINT_URL_BEDROCK_AGENT g:
Bedrock Agent Runtime	b: AWS_ENDPOINT_URL_BEDROCK_AGENT_RUNTIME g: ir
Bedrock Runtime	b: AWS_ENDPOINT_URL_BEDROCK_RUNTIME ur
billingconductor	b: AWS_ENDPOINT_URL_BILLINGCONDUCTOR n:
Braket	b: AWS_ENDPOINT_URL_BRAKET

serviceId	Cl	variable d'environnement AWS_ENDPOINT_URL_< d' SERVICE>
Budgets	bl	AWS_ENDPOINT_URL_BUDGETS
Cost Explorer	ce	AWS_ENDPOINT_URL_COST_EXPLORER
chatbot	cl	AWS_ENDPOINT_URL_CHATBOT
Chime	cl	AWS_ENDPOINT_URL_CHIME
Chime SDK Identity	cl	AWS_ENDPOINT_URL_CHIME_SDK_IDENTITY
Chime SDK Media Pipelines	cl	AWS_ENDPOINT_URL_CHIME_SDK_MEDIA_PIPELINES
Chime SDK Meetings	cl	AWS_ENDPOINT_URL_CHIME_SDK_MEETINGS
Chime SDK Messaging	cl	AWS_ENDPOINT_URL_CHIME_SDK_MESSAGING

serviceId	Cl
	variable d'environnement AWS_ENDPOINT_URL_< SERVICE>
Chime SDK Voice	cl AWS_ENDPOINT_URL_CHIME_SDK_VOICE
CleanRooms	c: AWS_ENDPOINT_URL_CLEANROOMS
CleanRoomsML	c: AWS_ENDPOINT_URL_CLEANROOMSML
Cloud9	c: AWS_ENDPOINT_URL_CLOUD9
CloudControl	c: AWS_ENDPOINT_URL_CLOUDCONTROL
CloudDirectory	c: AWS_ENDPOINT_URL_CLOUDDIRECTORY
CloudFormation	c: AWS_ENDPOINT_URL_CLOUDFORMATION
CloudFront	c: AWS_ENDPOINT_URL_CLOUDFRONT
CloudFront KeyValueStore	c: AWS_ENDPOINT_URL_CLOUDFRONT_KEYVALUESTORE

serviceId	Cl variable d'environnement AWS_ENDPOINT_URL_< d' SERVICE>
CloudHSM	c: AWS_ENDPOINT_URL_CLOUDHSM
CloudHSM V2	c: AWS_ENDPOINT_URL_CLOUDHSM_V2 v:
CloudSearch	c: AWS_ENDPOINT_URL_CLOUDSEARCH cl
CloudSearch Domain	c: AWS_ENDPOINT_URL_CLOUDSEARCH_DOMAIN cl
CloudTrail	c: AWS_ENDPOINT_URL_CLOUDTRAIL l
CloudTrail Data	c: AWS_ENDPOINT_URL_CLOUDTRAIL_DATA l_
CloudWatch	c: AWS_ENDPOINT_URL_CLOUDWATCH h
codeartifact	c: AWS_ENDPOINT_URL_CODEARTIFACT a
CodeBuild	c: AWS_ENDPOINT_URL_CODEBUILD

serviceId	Cl	variable d'environnement AWS_ENDPOINT_URL_< d' SERVICE>
CodeCatalyst	cc	AWS_ENDPOINT_URL_CODECATALYST
CodeCommit	cc	AWS_ENDPOINT_URL_CODECOMMIT
CodeDeploy	cc	AWS_ENDPOINT_URL_CODEDEPLOY
CodeGuru Reviewer	cc	AWS_ENDPOINT_URL_CODEGURU_REVIEWER
CodeGuru Security	cc	AWS_ENDPOINT_URL_CODEGURU_SECURITY
CodeGuruProfiler	cc	AWS_ENDPOINT_URL_CODEGURUPROFILER
CodePipeline	cc	AWS_ENDPOINT_URL_CODEPIPELINE
CodeStar	cc	AWS_ENDPOINT_URL_CODESTAR
CodeStar connections	cc	AWS_ENDPOINT_URL_CODESTAR_CONNECTIONS

serviceId	Cl variable d'environnement AWS_ENDPOINT_URL_< d' SERVICE>
codestar notifiations	cc AWS_ENDPOINT_URL_CODESTAR_NOTIFICATIONS no ic
Cognito Identity	cc AWS_ENDPOINT_URL_COGNITO_IDENTITY de
Cognito Identity Provider	cc AWS_ENDPOINT_URL_COGNITO_IDENTITY_PROVIDER de ic
Cognito Sync	cc AWS_ENDPOINT_URL_COGNITO_SYNC yl
Comprehend	cc AWS_ENDPOINT_URL_COMPREHEND d
ComprehendMedical	cc AWS_ENDPOINT_URL_COMPREHENDMEDICAL dr
Compute Optimizer	cc AWS_ENDPOINT_URL_COMPUTE_OPTIMIZER pl
Config Service	cc AWS_ENDPOINT_URL_CONFIG_SERVICE r\

serviceId	Cl
	variable d'environnement AWS_ENDPOINT_URL_< d' SERVICE>
Connect	co AWS_ENDPOINT_URL_CONNECT
Connect Contact Lens	co AWS_ENDPOINT_URL_CONNECT_CONTACT_LENS on n:
ConnectCampaigns	co AWS_ENDPOINT_URL_CONNECTCAMPAIGNS mp
ConnectCases	co AWS_ENDPOINT_URL_CONNECTCASES Se
ConnectParticipant	co AWS_ENDPOINT_URL_CONNECTPARTICIPANT It
ControlTower	co AWS_ENDPOINT_URL_CONTROLTOWER wé
Cost Optimization Hub	co AWS_ENDPOINT_URL_COST_OPTIMIZATION_HUB m: hu

serviceId	Cl
	variable d'environnement AWS_ENDPOINT_URL_< SERVICE>
Cost and Usage Report Service	cost_aws_endpoint_url_cost_and_usage_report_service
Customer Profiles	customer_profiles_aws_endpoint_url_customer_profiles
DataBrew	data_brew_aws_endpoint_url_databrew
DataExchange	data_exchange_aws_endpoint_url_dataexchange
Data Pipeline	data_pipeline_aws_endpoint_url_data_pipeline
DataSync	data_sync_aws_endpoint_url_datasync
DataZone	data_zone_aws_endpoint_url_datazone
DAX	dax_aws_endpoint_url_dax
Detective	detective_aws_endpoint_url_detective
Device Farm	device_farm_aws_endpoint_url_device_farm

serviceId	Cl	variable d'environnement AWS_ENDPOINT_URL_< d' SERVICE>
DevOps Guru	d:	AWS_ENDPOINT_URL_DEVOPS_GURU
Direct Connect	d:	AWS_ENDPOINT_URL_DIRECT_CONNECT
Application Discovery Service	a:	AWS_ENDPOINT_URL_APPLICATION_DISCOVER
	o:	RY_SERVICE
DLM	d:	AWS_ENDPOINT_URL_DLM
Database Migration Service	d:	AWS_ENDPOINT_URL_DATABASE_MIGRATION_
	m:	SERVICE
DocDB	d:	AWS_ENDPOINT_URL_DOCDB
DocDB Elastic	d:	AWS_ENDPOINT_URL_DOCDB_ELASTIC
drs	d:	AWS_ENDPOINT_URL_DRS
Directory Service	d:	AWS_ENDPOINT_URL_DIRECTORY_SERVICE

serviceId	Cl	variable d'environnement AWS_ENDPOINT_URL_< d' SERVICE>
DynamoDB	dy	AWS_ENDPOINT_URL_DYNAMODB
DynamoDB Streams	dy	AWS_ENDPOINT_URL_DYNAMODB_STREAMS
EBS	el	AWS_ENDPOINT_URL_EBS
EC2	ec	AWS_ENDPOINT_URL_EC2
EC2 Instance Connect	ec	AWS_ENDPOINT_URL_EC2_INSTANCE_CONNECT
ECR	ec	AWS_ENDPOINT_URL_ECR
ECR PUBLIC	ec	AWS_ENDPOINT_URL_ECR_PUBLIC
ECS	ec	AWS_ENDPOINT_URL_ECS
EFS	ef	AWS_ENDPOINT_URL_EFS
EKS	el	AWS_ENDPOINT_URL_EKS
EKS Auth	el	AWS_ENDPOINT_URL_EKS_AUTH

serviceId	Cl
	variable d'environnement AWS_ENDPOINT_URL_< SERVICE>
Elastic Inference	e: AWS_ENDPOINT_URL_ELASTIC_INFERENCE n:
ElastiCache	e: AWS_ENDPOINT_URL_ELASTICACHE h:
Elastic Beanstalk	e: AWS_ENDPOINT_URL_ELASTIC_BEANSTALK e:
Elastic Transcoder	e: AWS_ENDPOINT_URL_ELASTIC_TRANSCODER r:
Elastic Load Balancing	e: AWS_ENDPOINT_URL_ELASTIC_LOAD_BALANCING o: c:
Elastic Load Balancing v2	e: AWS_ENDPOINT_URL_ELASTIC_LOAD_BALANCING_V2 o: c:
EMR	er: AWS_ENDPOINT_URL_EMR
EMR containers	er: AWS_ENDPOINT_URL_EMR_CONTAINERS i:

serviceId	Cl
	variable d'environnement AWS_ENDPOINT_URL_< SERVICE>
EMR Serverless	er AWS_ENDPOINT_URL_EMR_SERVERLESS
EntityResolution	er AWS_ENDPOINT_URL_ENTITYRESOLUTION
Elasticsearch Service	e: AWS_ENDPOINT_URL_ELASTICSEARCH_SERVICE
EventBridge	ev AWS_ENDPOINT_URL_EVENTBRIDGE
Evidently	ev AWS_ENDPOINT_URL_EVIDENTLY
finspace	f: AWS_ENDPOINT_URL_FINSPACE
finspace data	f: AWS_ENDPOINT_URL_FINSPACE_DATA
Firehose	f: AWS_ENDPOINT_URL_FIREHOSE
fis	f: AWS_ENDPOINT_URL_FIS
FMS	fr AWS_ENDPOINT_URL_FMS

serviceId	Clé de la variable d'environnement AWS_ENDPOINT_URL_<SERVICE>
forecast	f: AWS_ENDPOINT_URL_FORECAST
forecastquery	f: AWS_ENDPOINT_URL_FORECASTQUERY
FraudDetector	f: AWS_ENDPOINT_URL_FRAUDETECTOR
FreeTier	f: AWS_ENDPOINT_URL_FREETIER
FSx	f: AWS_ENDPOINT_URL_FSX
GameLift	g: AWS_ENDPOINT_URL_GAMELIFT
Glacier	g: AWS_ENDPOINT_URL_GLACIER
Global Accelerator	g: AWS_ENDPOINT_URL_GLOBAL_ACCELERATOR
Glue	g: AWS_ENDPOINT_URL_GLUE
grafana	g: AWS_ENDPOINT_URL_GRAFANA
Greengrass	g: AWS_ENDPOINT_URL_GREENGRASS

serviceId	Cl variable d'environnement AWS_ENDPOINT_URL_< d' SERVICE>
GreengrassV2	g: AWS_ENDPOINT_URL_GREENGRASSV2 S\
GroundStation	g: AWS_ENDPOINT_URL_GROUNDSTATION t:
GuardDuty	g: AWS_ENDPOINT_URL_GUARDDUTY
Health	h: AWS_ENDPOINT_URL_HEALTH
HealthLake	h: AWS_ENDPOINT_URL_HEALTHLAKE e
Honeycode	h: AWS_ENDPOINT_URL_HONEYCODE
IAM	i: AWS_ENDPOINT_URL_IAM
identitystore	i: AWS_ENDPOINT_URL_IDENTITYSTORE t:
imagebuilder	i: AWS_ENDPOINT_URL_IMAGEBUILDER d:
ImportExport	i: AWS_ENDPOINT_URL_IMPORTEXPORT o:

serviceId	Cl
	variable d'environnement AWS_ENDPOINT_URL_< SERVICE>
Inspector	in AWS_ENDPOINT_URL_INSPECTOR
Inspector Scan	in AWS_ENDPOINT_URL_INSPECTOR_SCAN_
Inspector2	in AWS_ENDPOINT_URL_INSPECTOR2
InternetMonitor	in AWS_ENDPOINT_URL_INTERNETMONITOR
IoT	in AWS_ENDPOINT_URL_IOT
IoT Data Plane	in AWS_ENDPOINT_URL_IOT_DATA_PLANE
IoT Jobs Data Plane	in AWS_ENDPOINT_URL_IOT_JOBS_DATA_PLANE
IoT 1Click Devices Service	in AWS_ENDPOINT_URL_IOT_1CLICK_DEVICES_

serviceId	Cl variable d'environnement AWS_ENDPOINT_URL_< d' SERVICE>
IoT 1Click Projects	i AWS_ENDPOINT_URL_IOT_1CLICK_PROJECTS k_ s
IoTAnalytics	i AWS_ENDPOINT_URL_IOTANALYTICS i
IotDeviceAdvisor	i AWS_ENDPOINT_URL_IOTDEVICEADVISOR a
IoT Events	i AWS_ENDPOINT_URL_IOT_EVENTS s
IoT Events Data	i AWS_ENDPOINT_URL_IOT_EVENTS_DATA S_
IoTFleetHub	i AWS_ENDPOINT_URL_IOTFLEETHUB ul
IoTFleetWise	i AWS_ENDPOINT_URL_IOTFLEETWISE i
IoTSecureTunneling	i AWS_ENDPOINT_URL_IOTSECURETUNNELING t

serviceId	Cl	variable d'environnement AWS_ENDPOINT_URL_< d' SERVICE>
IoTSiteWise	i	AWS_ENDPOINT_URL_IOTSITEWISE
IoTThingsGraph	i	AWS_ENDPOINT_URL_IOTTHINGSGRAPH
IoTTwinMaker	i	AWS_ENDPOINT_URL_IOTTWINMAKER
IoT Wireless	i	AWS_ENDPOINT_URL_IOT_WIRELESS
ivs	i	AWS_ENDPOINT_URL_IVS
IVS RealTime	i	AWS_ENDPOINT_URL_IVS_REALTIME
ivschat	i	AWS_ENDPOINT_URL_IVSCHAT
Kafka	k	AWS_ENDPOINT_URL_KAFKA
KafkaConnect	k	AWS_ENDPOINT_URL_KAFKACONNECT
kendra	k	AWS_ENDPOINT_URL_KENDRA

serviceId	Cl	variable d'environnement AWS_ENDPOINT_URL_< d' SERVICE>
Kendra Ranking	k	AWS_ENDPOINT_URL_KENDRA_RANKING
Keyspaces	k	AWS_ENDPOINT_URL_KEYSPACES
Kinesis	k	AWS_ENDPOINT_URL_KINESIS
Kinesis Video Archived Media	k	AWS_ENDPOINT_URL_KINESIS_VIDEO_ARCHIVED_MEDIA
Kinesis Video Media	k	AWS_ENDPOINT_URL_KINESIS_VIDEO_MEDIA
Kinesis Video Signaling	k	AWS_ENDPOINT_URL_KINESIS_VIDEO_SIGNALING
Kinesis Video WebRTC Storage	k	AWS_ENDPOINT_URL_KINESIS_VIDEO_WEBRTC_STORAGE

serviceId	Cl
	variable d'environnement AWS_ENDPOINT_URL_< SERVICE>
Kinesis Analytics	k: AWS_ENDPOINT_URL_KINESIS_ANALYTICS
Kinesis Analytics V2	k: AWS_ENDPOINT_URL_KINESIS_ANALYTICS_V2
Kinesis Video	k: AWS_ENDPOINT_URL_KINESIS_VIDEO
KMS	kr: AWS_ENDPOINT_URL_KMS
LakeFormation	l: AWS_ENDPOINT_URL_LAKEFORMATION
Lambda	l: AWS_ENDPOINT_URL_LAMBDA
Launch Wizard	l: AWS_ENDPOINT_URL_LAUNCH_WIZARD
Lex Model Building Service	l: AWS_ENDPOINT_URL_LEX_MODEL_BUILDING_SERVICE

serviceId	Clé de variable d'environnement AWS_ENDPOINT_URL_<SERVICE>
Lex Runtime Service	AWS_ENDPOINT_URL_LEX_RUNTIME_SERVICE
Lex Models V2	AWS_ENDPOINT_URL_LEX_MODELS_V2
Lex Runtime V2	AWS_ENDPOINT_URL_LEX_RUNTIME_V2
License Manager	AWS_ENDPOINT_URL_LICENSE_MANAGER
License Manager Linux Subscriptions	AWS_ENDPOINT_URL_LICENSE_MANAGER_LINUX_SUBSCRIPTIONS
License Manager User Subscriptions	AWS_ENDPOINT_URL_LICENSE_MANAGER_USER_SUBSCRIPTIONS
Lightsail	AWS_ENDPOINT_URL_LIGHTSAIL

serviceId	Cl
	variable d'environnement AWS_ENDPOINT_URL_< d' SERVICE>
Location	l AWS_ENDPOINT_URL_LOCATION
CloudWatch Logs	c: AWS_ENDPOINT_URL_CLOUDWATCH_LOGS h_
CloudWatch Logs	c: AWS_ENDPOINT_URL_CLOUDWATCH_LOGS h_
LookoutEquipment	l AWS_ENDPOINT_URL_LOOKOUTEQUIPMENT u:
LookoutMetrics	l AWS_ENDPOINT_URL_LOOKOUTMETRICS t:
LookoutVision	l AWS_ENDPOINT_URL_LOOKOUTVISION s:
m2	m: AWS_ENDPOINT_URL_M2
Machine Learning	m: AWS_ENDPOINT_URL_MACHINE_LEARNING e:
Macie2	m: AWS_ENDPOINT_URL_MACIE2
ManagedBlockchain	m: AWS_ENDPOINT_URL_MANAGEDBLOCKCHAIN o:

serviceId	Cl variable d'environnement AWS_ENDPOINT_URL_< d' SERVICE>
ManagedBlockchain Query	m: AWS_ENDPOINT_URL_MANAGEDBLOCKCHAIN_QUERY o: q:
Marketplace Agreement	m: AWS_ENDPOINT_URL_MARKETPLACE_AGREEMENT c: e:
Marketplace Catalog	m: AWS_ENDPOINT_URL_MARKETPLACE_CATALOG c: g:
Marketplace Deployment	m: AWS_ENDPOINT_URL_MARKETPLACE_DEPLOYMENT c: m:
Marketplace Entitlement Service	m: AWS_ENDPOINT_URL_MARKETPLACE_ENTITLE c: MENT_SERVICE e: v:
Marketplace Commerce Analytics	m: AWS_ENDPOINT_URL_MARKETPLACE_COMMERC c: E_ANALYTICS c: i:

serviceId	Clé de variable d'environnement AWS_ENDPOINT_URL_<SERVICE>
MediaConnect	<code>AWS_ENDPOINT_URL_MEDIACONNECT</code>
MediaConvert	<code>AWS_ENDPOINT_URL_MEDIACONVERT</code>
MediaLive	<code>AWS_ENDPOINT_URL_MEDIALIVE</code>
MediaPackage	<code>AWS_ENDPOINT_URL_MEDIAPACKAGE</code>
MediaPackage Vod	<code>AWS_ENDPOINT_URL_MEDIAPACKAGE_VOD</code>
MediaPackageV2	<code>AWS_ENDPOINT_URL_MEDIAPACKAGEV2</code>
MediaStore	<code>AWS_ENDPOINT_URL_MEDIASTORE</code>
MediaStore Data	<code>AWS_ENDPOINT_URL_MEDIASTORE_DATA</code>
MediaTailor	<code>AWS_ENDPOINT_URL_MEDIATAILOR</code>

serviceId	Cl
	variable d'environnement AWS_ENDPOINT_URL_< SERVICE>
Medical Imaging	m: AWS_ENDPOINT_URL_MEDICAL_IMAGING
MemoryDB	m: AWS_ENDPOINT_URL_MEMORYDB
Marketplace Metering	m: AWS_ENDPOINT_URL_MARKETPLACE_METERING
Migration Hub	m: AWS_ENDPOINT_URL_MIGRATION_HUB
mgn	m: AWS_ENDPOINT_URL_MGN
Migration Hub Refactor Spaces	m: AWS_ENDPOINT_URL_MIGRATION_HUB_REFACTOR_SPACES
MigrationHub Config	m: AWS_ENDPOINT_URL_MIGRATIONHUB_CONFIG
MigrationHubOrchestrator	m: AWS_ENDPOINT_URL_MIGRATIONHUBORCHESTRATOR

serviceId	Clé de la variable d'environnement AWS_ENDPOINT_URL_<SERVICE>
MigrationHubStrategy	m: AWS_ENDPOINT_URL_MIGRATIONHUBSTRATEGY
Mobile	mo: AWS_ENDPOINT_URL_MOBILE
mq	mq: AWS_ENDPOINT_URL_MQ
MTurk	mt: AWS_ENDPOINT_URL_MTURK
MWAA	mw: AWS_ENDPOINT_URL_MWAA
Neptune	ne: AWS_ENDPOINT_URL_NEPTUNE
Neptune Graph	ng: AWS_ENDPOINT_URL_NEPTUNE_GRAPH
neptunedata	nd: AWS_ENDPOINT_URL_NEPTUNEDATA
Network Firewall	nf: AWS_ENDPOINT_URL_NETWORK_FIREWALL
NetworkManager	nm: AWS_ENDPOINT_URL_NETWORKMANAGER

serviceId	Clé de variable d'environnement AWS_ENDPOINT_URL_< SERVICE>
NetworkMonitor	networkmonitor: AWS_ENDPOINT_URL_NETWORKMONITOR
nimble	nimble: AWS_ENDPOINT_URL_NIMBLE
OAM	oam: AWS_ENDPOINT_URL_OAM
Omics	omics: AWS_ENDPOINT_URL_OMICS
OpenSearch	opensearch: AWS_ENDPOINT_URL_OPENSEARCH
OpenSearchServerless	opensearchserverless: AWS_ENDPOINT_URL_OPENSEARCHSERVERLESS
OpsWorks	opsworks: AWS_ENDPOINT_URL_OPSWORKS
OpsWorksCM	opsworkscm: AWS_ENDPOINT_URL_OPSWORKSCM
Organizations	organizations: AWS_ENDPOINT_URL_ORGANIZATIONS
OSIS	osis: AWS_ENDPOINT_URL_OSIS
Outposts	outposts: AWS_ENDPOINT_URL_OUTPOSTS

serviceId	Cl variable d'environnement AWS_ENDPOINT_URL_< d' SERVICE>
p8data	p AWS_ENDPOINT_URL_P8DATA
p8data	p AWS_ENDPOINT_URL_P8DATA
Panorama	p AWS_ENDPOINT_URL_PANORAMA
Payment Cryptography	p AWS_ENDPOINT_URL_PAYMENT_CRYPTOGRAPHY r h
Payment Cryptography Data	p AWS_ENDPOINT_URL_PAYMENT_CRYPTOGRAPHY_DATA r h
Pca Connector Ad	p AWS_ENDPOINT_URL_PCA_CONNECTOR_AD c
Personalize	p AWS_ENDPOINT_URL_PERSONALIZE z
Personalize Events	p AWS_ENDPOINT_URL_PERSONALIZE_EVENTS z

serviceId	Clé de variable d'environnement AWS_ENDPOINT_URL_<SERVICE>
Personalize Runtime	p: AWS_ENDPOINT_URL_PERSONALIZE_RUNTIME
PI	p: AWS_ENDPOINT_URL_PI
Pinpoint	p: AWS_ENDPOINT_URL_PINPOINT
Pinpoint Email	p: AWS_ENDPOINT_URL_PINPOINT_EMAIL
Pinpoint SMS Voice	p: AWS_ENDPOINT_URL_PINPOINT_SMS_VOICE
Pinpoint SMS Voice V2	p: AWS_ENDPOINT_URL_PINPOINT_SMS_VOICE_V2
Pipes	p: AWS_ENDPOINT_URL_PIPES
Polly	p: AWS_ENDPOINT_URL_POLLY
Pricing	p: AWS_ENDPOINT_URL_PRICING
PrivateNetworks	p: AWS_ENDPOINT_URL_PRIVATENETWORKS

serviceId	Clé de variable d'environnement AWS_ENDPOINT_URL_<SERVICE>
Proton	proton AWS_ENDPOINT_URL_PROTON
QBusiness	qbusiness AWS_ENDPOINT_URL_QBUSINESS
QConnect	qconnect AWS_ENDPOINT_URL_QCONNECT
QLDB	qldb AWS_ENDPOINT_URL_QLDB
QLDB Session	qldb-session AWS_ENDPOINT_URL_QLDB_SESSION
QuickSight	quicksight AWS_ENDPOINT_URL_QUICKSIGHT
RAM	ram AWS_ENDPOINT_URL_RAM
rbin	rbin AWS_ENDPOINT_URL_RBIN
RDS	rds AWS_ENDPOINT_URL_RDS
RDS Data	rds-data AWS_ENDPOINT_URL_RDS_DATA
Redshift	redshift AWS_ENDPOINT_URL_REDSHIFT
Redshift Data	redshift-data AWS_ENDPOINT_URL_REDSHIFT_DATA

serviceId	Clé de la variable d'environnement AWS_ENDPOINT_URL_<SERVICE>
Redshift Serverless	<code>AWS_ENDPOINT_URL_REDSHIFT_SERVERLESS</code>
Rekognition	<code>AWS_ENDPOINT_URL_REKOGNITION</code>
repostspace	<code>AWS_ENDPOINT_URL_REPOSTSPACE</code>
resiliencehub	<code>AWS_ENDPOINT_URL_RESILIENCEHUB</code>
Resource Explorer 2	<code>AWS_ENDPOINT_URL_RESOURCE_EXPLORER_2</code>
Resource Groups	<code>AWS_ENDPOINT_URL_RESOURCE_GROUPS</code>
Resource Groups Tagging API	<code>AWS_ENDPOINT_URL_RESOURCE_GROUPS_TAGGING_API</code>

serviceId	Cl
	variable d'environnement AWS_ENDPOINT_URL_< SERVICE>
RoboMaker	<code>AWS_ENDPOINT_URL_ROBOMAKER</code>
RolesAnywhere	<code>AWS_ENDPOINT_URL_ROLESEANYWHERE</code>
Route 53	<code>AWS_ENDPOINT_URL_ROUTE_53</code>
Route53 Recovery Cluster	<code>AWS_ENDPOINT_URL_ROUTE53_RECOVERY_CLUSTER</code>
Route53 Recovery Control Config	<code>AWS_ENDPOINT_URL_ROUTE53_RECOVERY_CONTROL_CONFIG</code>
Route53 Recovery Readiness	<code>AWS_ENDPOINT_URL_ROUTE53_RECOVERY_READINESS</code>
Route 53 Domains	<code>AWS_ENDPOINT_URL_ROUTE_53_DOMAINS</code>
Route53Resolver	<code>AWS_ENDPOINT_URL_ROUTE53RESOLVER</code>

serviceId	Clé de la variable d'environnement AWS_ENDPOINT_URL_< SERVICE>
RUM	AWSEndpointUrlRum
S3	AWSEndpointUrlS3
S3 Control	AWSEndpointUrlS3Control
S3Outposts	AWSEndpointUrlS3Outposts
SageMaker	AWSEndpointUrlSageMaker
SageMaker A2I Runtime	AWSEndpointUrlSageMakerA2IRuntime
Sagemaker Edge	AWSEndpointUrlSageMakerEdge
SageMaker FeatureStore Runtime	AWSEndpointUrlSageMakerFeatureStoreRuntime

serviceId	Cl
	variable d'environnement AWS_ENDPOINT_URL_< SERVICE>
SageMaker Geospatial	s: AWS_ENDPOINT_URL_SAGEMAKER_GEOSPATIAL
SageMaker Metrics	s: AWS_ENDPOINT_URL_SAGEMAKER_METRICS
SageMaker Runtime	s: AWS_ENDPOINT_URL_SAGEMAKER_RUNTIME
savingsplans	s: AWS_ENDPOINT_URL_SAVINGSPLANS
Scheduler	s: AWS_ENDPOINT_URL_SCHEDULER
schemas	s: AWS_ENDPOINT_URL_SCHEMAS
SimpleDB	s: AWS_ENDPOINT_URL_SIMPLEDB
Secrets Manager	s: AWS_ENDPOINT_URL_SECRETS_MANAGER
SecurityHub	s: AWS_ENDPOINT_URL_SECURITYHUB

serviceId	Cl	variable d'environnement AWS_ENDPOINT_URL_< d' SERVICE>
SecurityLake	se	AWS_ENDPOINT_URL_SECURITYLAKE
ServerlessApplicationRepository	se	AWS_ENDPOINT_URL_SERVERLESSAPPLICATI
Service Quotas	se	AWS_ENDPOINT_URL_SERVICE_QUOTAS
Service Catalog	se	AWS_ENDPOINT_URL_SERVICE_CATALOG
Service Catalog AppRegistry	se	AWS_ENDPOINT_URL_SERVICE_CATALOG_APP
ServiceDiscovery	se	AWS_ENDPOINT_URL_SERVICEDISCOVERY
SES	se	AWS_ENDPOINT_URL_SES
SESV2	se	AWS_ENDPOINT_URL_SESV2

serviceId	Cl	variable d'environnement AWS_ENDPOINT_URL_< d' SERVICE>
Shield	si	AWS_ENDPOINT_URL_SHIELD
signer	s:	AWS_ENDPOINT_URL_SIGNER
SimSpaceWeaver	s:	AWS_ENDPOINT_URL_SIMSPACEWEAVER
SMS	si	AWS_ENDPOINT_URL_SMS
Snow Device Management	si	AWS_ENDPOINT_URL_SNOW_DEVICE_MANAGEMENT
Snowball	si	AWS_ENDPOINT_URL_SNOWBALL
SNS	si	AWS_ENDPOINT_URL_SNS
SQS	si	AWS_ENDPOINT_URL_SQS
SSM	s:	AWS_ENDPOINT_URL_SSM
SSM Contacts	s:	AWS_ENDPOINT_URL_SSM_CONTACTS
SSM Incidents	s:	AWS_ENDPOINT_URL_SSM_INCIDENTS

serviceId	Cl	variable d'environnement AWS_ENDPOINT_URL_< d' SERVICE>
Ssm Sap	s:	AWS_ENDPOINT_URL_SSM_SAP
SSO	s:	AWS_ENDPOINT_URL_SSO
SSO Admin	s:	AWS_ENDPOINT_URL_SSO_ADMIN
SSO OIDC	s:	AWS_ENDPOINT_URL_SSO_OIDC
SFN	s:	AWS_ENDPOINT_URL_SFN
Storage Gateway	s:	AWS_ENDPOINT_URL_STORAGE_GATEWAY
STS	s:	AWS_ENDPOINT_URL_STS
SupplyChain	s:	AWS_ENDPOINT_URL_SUPPLYCHAIN
Support	s:	AWS_ENDPOINT_URL_SUPPORT
Support App	s:	AWS_ENDPOINT_URL_SUPPORT_APP
SWF	s:	AWS_ENDPOINT_URL_SWF

serviceId	Cl variable d'environnement AWS_ENDPOINT_URL_< d' SERVICE> ic: de se pc Al co le fic pa
synthetics	sy AWS_ENDPOINT_URL_SYNTHETICS s
Textract	te AWS_ENDPOINT_URL_TEXTRACT
Timestream InfluxDB	t: AWS_ENDPOINT_URL_TIMESTREAM_INFLUXDB m_ b
Timestream Query	t: AWS_ENDPOINT_URL_TIMESTREAM_QUERY m_
Timestream Write	t: AWS_ENDPOINT_URL_TIMESTREAM_WRITE m_
tnb	tr AWS_ENDPOINT_URL_TNB
Transcribe	t: AWS_ENDPOINT_URL_TRANSCRIBE e
Transfer	t: AWS_ENDPOINT_URL_TRANSFER
Translate	t: AWS_ENDPOINT_URL_TRANSLATE
TrustedAdvisor	t: AWS_ENDPOINT_URL_TRUSTEDADVISOR v:

serviceId	Cl variable d'environnement AWS_ENDPOINT_URL_< d' SERVICE> ic: de se pc Al co le fic pa
VerifiedPermissions	ve AWS_ENDPOINT_URL_VERIFIEDPERMISSIONS e: s
Voice ID	vo AWS_ENDPOINT_URL_VOICE_ID
VPC Lattice	vp AWS_ENDPOINT_URL_VPC_LATTICE Cé
WAF	wi AWS_ENDPOINT_URL_WAF
WAF Regional	wi AWS_ENDPOINT_URL_WAF_REGIONAL ni
WAFV2	wi AWS_ENDPOINT_URL_WAFV2
WellArchitected	wi AWS_ENDPOINT_URL_WELLARCHITECTED te
Wisdom	wi AWS_ENDPOINT_URL_WISDOM
WorkDocs	wi AWS_ENDPOINT_URL_WORKDOCS
WorkLink	wi AWS_ENDPOINT_URL_WORKLINK
WorkMail	wi AWS_ENDPOINT_URL_WORKMAIL

serviceId	Cl variable d'environnement AWS_ENDPOINT_URL_< d' SERVICE>
WorkMailMessageFlow	w AWS_ENDPOINT_URL_WORKMAILMESSAGEFLOW e: w
WorkSpaces	w AWS_ENDPOINT_URL_WORKSPACES S
WorkSpaces Thin Client	w AWS_ENDPOINT_URL_WORKSPACES_THIN_CLIENT S_ i
WorkSpaces Web	w AWS_ENDPOINT_URL_WORKSPACES_WEB S_
XRay	x: AWS_ENDPOINT_URL_XRAY

Paramètres de configuration intelligents par défaut

Grâce à la fonctionnalité de configuration intelligente par défaut, AWS les SDK peuvent fournir des valeurs par défaut prédéfinies et optimisées pour d'autres paramètres de configuration.

Configurez cette fonctionnalité à l'aide des méthodes suivantes :

defaults_mode- réglage AWS **config** du fichier partagé, **AWS_DEFAULTS_MODE**- variable d'environnement, **aws.defaultsMode**- Propriété du système JVM : Java/Kotlin uniquement

Avec ce paramètre, vous pouvez choisir un mode qui s'aligne sur l'architecture de votre application, qui fournit ensuite des valeurs par défaut optimisées pour votre application. Si une valeur est explicitement définie pour un paramètre du AWS SDK, cette valeur est toujours prioritaire. Si aucune valeur n'est définie explicitement pour un paramètre du AWS SDK et qu'`defaults_mode` n'est pas égal à l'ancien paramètre, cette fonctionnalité peut fournir des valeurs par défaut différentes pour différents paramètres optimisés pour votre application. Les paramètres peuvent inclure les éléments suivants : les paramètres de communication HTTP, le comportement des nouvelles tentatives, les paramètres du point de terminaison régional du service et, éventuellement, toute configuration liée au SDK. Les clients qui utilisent cette fonctionnalité peuvent obtenir de nouvelles configurations par défaut adaptées aux scénarios d'utilisation courants. Si vous `defaults_mode` n'êtes pas égal à `legacy`, nous vous recommandons de tester votre application lors de la mise à niveau du SDK, car les valeurs par défaut fournies peuvent changer en fonction de l'évolution des meilleures pratiques.

Valeur par défaut : `legacy`

Remarque : les nouvelles versions majeures des SDK seront définies par défaut sur `standard`.

Valeurs valides:

- `legacy`— Fournit des paramètres par défaut qui varient selon le SDK et qui existaient avant la création de `defaults_mode`.
- `standard`— Fournit les dernières valeurs par défaut recommandées qui devraient pouvoir être exécutées en toute sécurité dans la plupart des scénarios.
- `in-region`— S'appuie sur le mode `standard` et inclut une optimisation adaptée aux applications qui appellent Services AWS depuis le même mode Région AWS.
- `cross-region`— S'appuie sur le mode `standard` et inclut une optimisation adaptée aux applications faisant appel Services AWS à une région différente.
- `mobile`— S'appuie sur le mode `standard` et inclut une optimisation adaptée aux applications mobiles.
- `auto`— S'appuie sur le mode `standard` et inclut des fonctionnalités expérimentales. Le SDK tente de découvrir l'environnement d'exécution afin de déterminer automatiquement les paramètres appropriés. La détection automatique est basée sur l'heuristique et ne fournit pas une précision de 100 %. Si l'environnement d'exécution ne peut pas être déterminé, `standard`

le mode est utilisé. La détection automatique peut interroger les [métadonnées de l'instance et les données utilisateur](#), ce qui peut introduire de la latence. Si la latence de démarrage est essentielle pour votre application, nous vous recommandons de choisir une latence explicite à la `defaults_mode` place.

Exemple de définition de cette valeur dans le config fichier :

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

Les paramètres suivants peuvent être optimisés en fonction de la sélection de `defaults_mode` :

- `retryMode`— Spécifie la manière dont le SDK tente de réessayer. veuillez consulter [Comportement de nouvelle tentative](#).
- `stsRegionalEndpoints`— Spécifie la manière dont le SDK détermine le Service AWS point de terminaison qu'il utilise pour communiquer avec le AWS Security Token Service (AWS STS). veuillez consulter [AWS STS Points de terminaison régionalisés](#).
- `s3UsEast1RegionalEndpoints`— Spécifie la manière dont le SDK détermine le point de terminaison du AWS service qu'il utilise pour communiquer avec Amazon S3 pour la `us-east-1` région.
- `connectTimeoutInMillis`— Après avoir effectué une première tentative de connexion sur un socket, délai avant l'expiration du délai imparti. Si le client ne reçoit pas de confirmation de connexion terminée, il abandonne l'opération et échoue.
- `tlsNegotiationTimeoutInMillis`— Le temps maximum qu'une poignée de main TLS peut prendre entre le moment où le message CLIENT HELLO est envoyé et le moment où le client et le serveur ont entièrement négocié les chiffrements et échangé les clés.

La valeur par défaut de chaque paramètre change en fonction de `defaults_mode` celui sélectionné pour votre application. Ces valeurs sont actuellement définies comme suit (sous réserve de modifications) :

Paramètre	Mode standard	Mode in-region	Mode cross-region	Mode mobile
<code>retryMode</code>	standard	standard	standard	standard

Paramètre	Mode standard	Mode in-region	Mode cross-region	Mode mobile
<code>stsRegionalEndpoints</code>	regional	regional	regional	regional
<code>s3UsEast1RegionalEndpoints</code>	regional	regional	regional	regional
<code>connectTimeoutInMillis</code>	3100	1100	3100	30 000
<code>tlsNegotiationTimeoutInMillis</code>	3100	1100	3100	30 000

Par exemple, si `defaults_mode` vous avez sélectionné l'état `standard`, la valeur de `standard` serait attribuée à `retry_mode` (à partir des `retry_mode` options valides) et la valeur de `regional` serait affectée à `stsRegionalEndpoints` (à partir des `stsRegionalEndpoints` options valides).

Compatibilité avec les AWS SDK

Les SDK suivants prennent en charge les fonctionnalités et les paramètres décrits dans cette rubrique. Toute exception partielle est notée. Tous les paramètres de propriété du système JVM sont pris en charge par le AWS SDK for Java et le Kit AWS SDK pour Kotlin seul.

Kit SDK	Pris en charge	Remarques ou informations supplémentaires
AWS CLI v2	Non	
SDK pour C++	Oui	Paramètres non optimisés : <code>stsRegion</code>

Kit SDK	Pris en charge	Remarques ou informations supplémentaires
		<code>alEndpoint</code> <code>ts ,s3UsEast1Regiona</code> <code>Endpoints ,tlsNegoti</code> <code>ationTimeoutInMill</code> <code>is .</code>
SDK pour Go V2 (1.x)	Oui	Paramètres non optimisés : <code>retryMode</code> <code>,stsRegionalEndpoin</code> <code>ts ,s3UsEast1Regiona</code> <code>Endpoints .</code>
SDK pour Go 1.x (V1)	Non	
SDK pour Java 2.x	Oui	Paramètres non optimisés : <code>stsRegion</code> <code>alEndpoints .</code>
SDK pour Java 1.x	Non	
SDK pour 3.x JavaScript	Oui	Paramètres non optimisés : <code>stsRegion</code> <code>alEndpoint</code> <code>ts ,s3UsEast1Regiona</code> <code>Endpoints ,tlsNegoti</code> <code>ationTimeoutInMill</code> <code>is . connectTi</code> <code>meoutInMillis est</code> <code>appeléconnectio</code> <code>nTimeout .</code>
SDK pour 2.x JavaScript	Non	
SDK pour Kotlin	Non	

Kit SDK	Pris en charge	Remarques ou informations supplémentaires
SDK pour .NET 3.x	Oui	Paramètres non optimisés :connectTimeoutInMillis ,tlsNegotiationTimeoutInMillis .
SDK pour PHP 3.x	Oui	Paramètres non optimisés :tlsNegotiationTimeoutInMillis .
SDK pour Python (Boto3)	Oui	Paramètres non optimisés :tlsNegotiationTimeoutInMillis .
SDK pour Ruby 3.x	Oui	
SDK pour Rust	Non	
Outils pour PowerShell	Oui	Paramètres non optimisés :connectTimeoutInMillis ,tlsNegotiationTimeoutInMillis .

AWSbibliothèques CRT (Common Runtime)

Les bibliothèques CRT (AWSCommon Runtime) sont une bibliothèque de base des SDK. Le CRT est une famille modulaire de packages indépendants, écrits en C. Chaque package fournit de bonnes performances et un encombrement minimal pour les différentes fonctionnalités requises. Ces fonctionnalités sont communes et partagées par tous les SDK, ce qui améliore la réutilisation, l'optimisation et la précision du code. Les packages sont les suivants :

- [awslabs/aws-c-auth](https://github.com/awslabs/aws-c-auth): authentification AWS côté client (fournisseurs d'informations d'identification standard et signature (sigv4))
- [awslabs/aws-c-cal](https://github.com/awslabs/aws-c-cal): types primitifs cryptographiques, hachages (MD5, SHA256, SHA256 HMAC), signataires, AES
- [awslabs/aws-c-common](https://github.com/awslabs/aws-c-common): structures de données de base, types primitifs de threading/synchronisation, gestion de la mémoire tampon, fonctions liées à stdlib
- [awslabs/aws-c-compression](https://github.com/awslabs/aws-c-compression): Algorithmes de compression (codage/décodage Huffman)
- [awslabs/aws-c-event-stream](https://github.com/awslabs/aws-c-event-stream): traitement des messages des flux d'événements (en-têtes, prélude, charge utile, crc/trailer), implémentation des appels de procédure à distance (RPC) sur les flux d'événements
- [awslabs/aws-c-http](https://github.com/awslabs/aws-c-http): Implémentation C99 des spécifications HTTP/1.1 et HTTP/2
- [awslabs/aws-c-io](https://github.com/awslabs/aws-c-io): sockets (TCP, UDP), DNS, canaux, boucles d'événements, canaux, SSL/TLS
- [awslabs/aws-c-iot](https://github.com/awslabs/aws-c-iot): C99 Mise en œuvre de l'intégration des services cloud AWS IoT aux appareils
- [awslabs/aws-c-mqtt](https://github.com/awslabs/aws-c-mqtt): protocole de messagerie standard et léger pour l'Internet des objets (IoT)
- [awslabs/aws-c-s3](https://github.com/awslabs/aws-c-s3): implémentation de la bibliothèque C99 pour communiquer avec le service Amazon S3, conçue pour optimiser le débit sur les instances Amazon EC2 à bande passante élevée
- [awslabs/aws-c-sdkutils](https://github.com/awslabs/aws-c-sdkutils): bibliothèque d'utilitaires pour l'analyse et la gestion des profils AWS
- [awslabs/aws-checksums](https://github.com/awslabs/aws-checksums): CRC32c et CRC32 multiplateformes accélérés par matériel avec repli sur des implémentations logicielles efficaces
- [awslabs/aws-1c](https://github.com/awslabs/aws-1c): bibliothèque cryptographique à usage général gérée par l'équipe de AWS cryptographie pour AWS et ses clients, basée sur le code du projet Google BoringSSL et du projet OpenSSL

- [aws1abs/s2n](#): Implémentation C99 des protocoles TLS/SSL, conçus pour être petits et rapides avec la sécurité comme priorité

Le CRT est disponible via tous les SDK à l'exception de Go.

Dépendances CRT

Les bibliothèques CRT forment un réseau complexe de relations et de dépendances. Connaître ces relations est utile si vous devez créer le CRT directement à partir de la source. Cependant, la plupart des utilisateurs accèdent aux fonctionnalités CRT par le biais de leur SDK linguistique (tel que le AWS SDK pour C++ ou le SDK pour AWS Java) ou du SDK de leur appareil IoT (tel que le SDK IoT pour AWS C++ ou le SDK IoT pour Java). AWS Dans le schéma suivant, la zone Language CRT Bindings fait référence au package qui contient les bibliothèques CRT pour un SDK de langage spécifique. Il s'agit d'une collection de packages du format `aws-crt-*`, où « * » est un langage du SDK (tel que [aws-crt-cpp](#) ou [aws-crt-java](#)).

Voici une illustration des dépendances hiérarchiques des bibliothèques CRT.

AWS Politique de maintenance des SDK et des outils

Présentation

Ce document décrit la politique de maintenance des kits de développement AWS logiciel (SDK) et des outils, y compris les SDK mobiles et IoT, ainsi que leurs dépendances sous-jacentes. AWS fournit régulièrement aux AWS SDK et aux outils des mises à jour qui peuvent inclure la prise en charge d'AWS API nouvelles ou mises à jour, de nouvelles fonctionnalités, des améliorations, des corrections de bogues, des correctifs de sécurité ou des mises à jour de documentation. Les mises à jour peuvent également prendre en compte les modifications liées aux dépendances, aux environnements d'exécution des langages et aux systèmes d'exploitation. AWS Les versions du SDK sont publiées dans les gestionnaires de packages (par exemple Maven, NuGet PyPI) et sont disponibles sous forme de code source sur. GitHub

Nous recommandons aux utilisateurs de s'en tenir up-to-date aux versions du SDK pour rester au fait des dernières fonctionnalités, des mises à jour de sécurité et des dépendances sous-jacentes. L'utilisation continue d'une version du SDK non prise en charge n'est pas recommandée et est laissée à la discrétion de l'utilisateur.

Gestion des versions

Les versions du AWS SDK se présentent sous la forme de X.Y.Z où X représente la version majeure. L'augmentation de la version principale d'un SDK indique que ce SDK a subi des modifications importantes et substantielles pour prendre en charge les nouvelles expressions idiomatiques et les nouveaux modèles du langage. Les versions majeures sont introduites lorsque les interfaces publiques (par exemple, les classes, les méthodes, les types, etc.), les comportements ou la sémantique ont changé. Les applications doivent être mises à jour pour fonctionner avec la dernière version du SDK. Il est important de mettre à jour les versions majeures avec soin et conformément aux directives de mise à niveau fournies par AWS.

Cycle de vie des versions majeures du SDK

Le cycle de vie des principales versions des SDK et des outils comprend 5 phases, décrites ci-dessous.

- Version préliminaire pour les développeurs (phase 0) - Au cours de cette phase, les SDK ne sont pas pris en charge, ne doivent pas être utilisés dans les environnements de production et

sont uniquement destinés à un accès anticipé et à des fins de commentaires. Il est possible que les futures versions introduisent des modifications majeures. Une fois AWS qu'une version a été identifiée comme étant un produit stable, elle peut la marquer comme Release Candidate. Les versions candidates sont prêtes pour la version GA à moins que des bogues importants ne surviennent, et bénéficieront d'un AWS support complet.

- Disponibilité générale (GA) (phase 1) - Au cours de cette phase, les SDK sont entièrement pris en charge. AWS fournira des versions régulières du SDK qui incluent la prise en charge de nouveaux services, des mises à jour d'API pour les services existants, ainsi que des correctifs de bogues et de sécurité. Pour les outils, AWS fournira des versions régulières incluant de nouvelles mises à jour de fonctionnalités et des corrections de bogues. AWS prendra en charge la version GA d'un SDK pendant au moins 24 mois.
- Annonce de maintenance (phase 2) : AWS fera une annonce publique au moins 6 mois avant qu'un SDK ne passe en mode maintenance. Pendant cette période, le SDK continuera d'être entièrement pris en charge. Généralement, le mode maintenance est annoncé au moment où la prochaine version majeure passe en GA.
- Maintenance (phase 3) - Pendant le mode maintenance, AWS limite les versions du SDK pour résoudre les corrections de bogues critiques et les problèmes de sécurité uniquement. Un SDK ne recevra pas de mises à jour d'API pour les services nouveaux ou existants, ni ne sera mis à jour pour prendre en charge de nouvelles régions. Le mode maintenance a une durée par défaut de 12 mois, sauf indication contraire.
- Fin du support (phase 4) - Lorsqu'un SDK atteint la fin du support, il ne reçoit plus de mises à jour ni de versions. Les versions précédemment publiées continueront d'être disponibles via les gestionnaires de packages publics et le code restera activé GitHub. Le GitHub référentiel peut être archivé. L'utilisation d'un SDK atteint end-of-support est laissée à la discrétion de l'utilisateur. Nous recommandons aux utilisateurs de passer à la nouvelle version majeure.

Voici une illustration visuelle du cycle de vie des versions majeures du SDK. Veuillez noter que les délais indiqués ci-dessous sont illustratifs et ne sont pas contraignants.

Cycle de vie des dépendances

La plupart des AWS SDK ont des dépendances sous-jacentes, telles que des environnements d'exécution de langage, des systèmes d'exploitation ou des bibliothèques et frameworks tiers. Ces dépendances sont généralement liées à la communauté linguistique ou au fournisseur propriétaire du composant en question. Chaque communauté ou fournisseur publie son propre end-of-support calendrier pour son produit.

Les termes suivants sont utilisés pour classer les dépendances tierces sous-jacentes :

- Système d'exploitation (OS) : les exemples incluent Amazon Linux AMI, Amazon Linux 2, Windows 2008, Windows 2012, Windows 2016, etc.
- Language Runtime : les exemples incluent Java 7, Java 8, Java 11, .NET Core, .NET Standard, .NET PCL, etc.
- Bibliothèque/framework tiers : les exemples incluent OpenSSL, .NET Framework 4.5, Java EE, etc.

Notre politique est de continuer à prendre en charge les dépendances du SDK pendant au moins 6 mois après que la communauté ou le fournisseur a mis fin au support de ces dépendances. Cette politique peut toutefois varier en fonction de la dépendance spécifique.

Note

AWS se réserve le droit de mettre fin à la prise en charge d'une dépendance sous-jacente sans augmenter la version majeure du SDK

Méthodes de communication

Les annonces de maintenance sont communiquées de plusieurs manières :

- Un e-mail d'annonce est envoyé aux comptes concernés, annonçant notre intention de mettre fin au support pour la version spécifique du SDK. L'e-mail indiquera le chemin à suivre end-of-support, précisera le calendrier de la campagne et fournira des conseils de mise à niveau.
- AWS La documentation du SDK, telle que la documentation de référence des API, les guides de l'utilisateur, les pages marketing des produits du SDK et les fichiers GitHub readme, est mise à jour pour indiquer le calendrier de la campagne et fournir des conseils sur la mise à niveau des applications concernées.
- Un article de AWS blog est publié qui décrit le chemin à end-of-support suivre et réitère le calendrier de la campagne.
- Des avertissements d'obsolescence sont ajoutés aux SDK, indiquant le chemin d'accès à la documentation du SDK end-of-support et renvoyant à celle-ci.

Pour consulter la liste des versions majeures disponibles des AWS SDK et des outils et leur état d'avancement dans leur cycle de maintenance, voir [Matrice de support des versions](#).

AWS Matrice de prise en charge des versions des SDK et des outils

La matrice ci-dessous présente la liste des versions majeures du kit de développement AWS logiciel (SDK) disponibles et indique où elles en sont dans le cycle de maintenance, avec les délais associés. Pour des informations détaillées sur le cycle de vie des versions principales des AWS SDK et des outils et sur leurs dépendances sous-jacentes, voir [Politique de maintenance](#).

SDK	Version majeure	Phase en cours	Date de disponibilité générale	Remarques
AWS CLI	1.x	Disponibilité générale	02/09/2013	
AWS CLI	2.x	Disponibilité générale	10/02/2020	
SDK pour C++	1.x	Disponibilité générale	02/09/2015	
SDK pour Go V2	V2.1.x	Disponibilité générale	19/01/2021	
SDK pour Go	1.x	Annonce de maintenance	19/11/2015	Voir l'annonce pour les détails et les dates
SDK pour Java	1.x	Annonce de maintenance	25/03/2010	Voir l'annonce pour les détails et les dates
SDK pour Java	2.x	Disponibilité générale	11-20-2018	
SDK pour JavaScript	1.x	Fin du support	06/05/2013	

SDK	Version majeure	Phase en cours	Date de disponibilité générale	Remarques
SDK pour JavaScript	2.x	Annonce de maintenance	19/06/2014	Voir l'annonce pour les détails et les dates
SDK pour JavaScript	3.x	Disponibilité générale	15 décembre 2020	
SDK pour Kotlin	1.x	Disponibilité générale	27/11/2023	
SDK pour .NET	1.x	Fin du support	11/2009	
SDK pour .NET	2.x	Fin du support	08/11/2013	
SDK pour .NET	3.x	Disponibilité générale	28/07/2015	
SDK pour PHP	2.x	Fin du support	02/11/2012	
SDK pour PHP	3.x	Disponibilité générale	27/05/2015	
SDK pour Python (Boto2)	1.x	Fin du support	13/07/2011	
SDK pour Python (Boto3)	1.x	Disponibilité générale	22/06/2015	
SDK pour Python (Botocore)	1.x	Disponibilité générale	22/06/2015	
SDK pour Ruby	1.x	Fin du support	14/07/2011	
SDK pour Ruby	2.x	Fin du support	15/02/2015	

SDK	Version majeure	Phase en cours	Date de disponibilité générale	Remarques
SDK pour Ruby	3.x	Disponibilité générale	29/08/2017	
SDK pour Rust	1.x	Disponibilité générale	27/11/2023	
SDK pour Swift	1.x	Version préliminaire destinée aux développeurs		
Outils pour PowerShell	2.x	Fin du support	08/11/2013	
Outils pour PowerShell	3.x	Fin du support	29/07/2015	
Outils pour PowerShell	4. x	Disponibilité générale	21/11/2019	

Guide de référence sur l'historique des documents AWS relatifs aux SDK et aux outils

Le tableau suivant décrit les ajouts et mises à jour importants apportés au Guide de référence AWS des SDK et des outils. Pour recevoir les notifications sur les mises à jour de cette documentation, vous pouvez vous abonner au Flux RSS.

Modification	Description	Date
Propriétés du système SDK for Java 1.x	Ajoutez des détails sur les paramètres de configuration du système JVM pris en charge par la version AWS SDK for Java 1.x.	30 mai 2024
Mise à jour des paramètres	Ajoutez les paramètres de configuration du système JVM.	27 mars 2024
Mise à jour du tableau de compatibilité	Mises à jour de compatibilité pour le support du SDK, mises à jour des procédures IAM Identity Center.	20 février 2024
Mise à jour des informations d'identification du conteneur. Mise à jour IMDS.	Ajout de la prise en charge d'Amazon EKS. Ajout d'un paramètre pour désactiver le repli IMDSv1.	29 décembre 2023
Compression des demandes	Ajout de paramètres pour la fonction de compression des demandes.	27 décembre 2023
Tableaux de compatibilité	Les tableaux de compatibilité pour les fonctionnalités du SDK et des outils ont été mis à jour pour inclure le SDK pour	10 décembre 2023

	Kotlin, le SDK pour Rust et. AWS Tools for PowerShell	
Mises à jour de	Mises à jour des méthodes d'authentification prises en charge pour les SDK et les outils.	1er juillet 2023
Mises à jour des bonnes pratiques IAM	Mise à jour du guide s'aligner sur les bonnes pratiques IAM. Pour plus d'informations, consultez Bonnes pratiques de sécurité dans IAM .	27 février 2023
Mises à jour du SSO	Mises à jour des informations d'identification SSO pour la nouvelle configuration du jeton SSO.	19 novembre 2022
Mise à jour des paramètres	Mises à jour du tableau de support pour la configura tion générale et pour les points d'accès multirégionaux Amazon S3.	17 novembre 2022
Mise à jour des paramètres	Mises à jour visant à clarifier les informations d'identification du client IMDS et de l'IMDS. Mises à jour des variables d'environnement.	4 novembre 2022
Mettre à jour la page d'accueil	Annnonce d'Amazon CodeWhisperer.	22 septembre 2022
Changement de nom de service pour l'authentification unique	Mises à jour pour refléter le fait que le AWS SSO est désormais appelé. AWS IAM Identity Center	26 juillet 2022

[Mise à jour des paramètres](#)

Mises à jour mineures des détails du fichier de configuration et des paramètres pris en charge.

15 juin 2022

[Mettre à jour](#)

Mise à jour massive de presque toutes les parties de ce guide.

1er février 2022

[Première version](#)

La première version de ce guide est mise à la disposition du public.

13 mars 2020

Glossaire AWS

Pour connaître la terminologie la plus récente d'AWS, consultez le [Glossaire AWS](#) dans la Référence Glossaire AWS.

Les traductions sont fournies par des outils de traduction automatique. En cas de conflit entre le contenu d'une traduction et celui de la version originale en anglais, la version anglaise prévaudra.