aws

参考指南

AWS SDKs和工具

AWS SDKs和工具: 参考指南

Table of Contents

AWS SDKs和《工具参考指南》

许多SDKs工具通过共享的设计规范或共享库共享一些共同的功能。

本指南包含有关以下内容的信息：

- 配置— 如何使用共享config和credentials文件或环境变量来配置 AWS SDKs和工具。
- 身份验证和访问— 确定您的代码或工具在使用开发 AWS 时如何进行身份验证。 AWS 服务
- 设置参考 – 所有可用于身份验证和配置的标准化设置的参考。
- AWS 常用运行时 (CRT) 库— 几乎所有人都可以使用的共享 AWS 通用运行时 (CRT) 库概述SDKs。
- AWS SDK 和工具维护政策涵盖 AWS 软件开发套件 (SDKs) 和工具（包括移动和物联网 (IoT)）SDKs的维护策略和版本控制及其底层依赖关系。

本 AWS SDKs和工具参考指南旨在成为适用于多种工具SDKs的信息库。除此处提供的任何信息外，还应使用您正在使用的SDK或工具的特定指南。以下是SDK和工具，其中包含本指南中相关材料的章节：

如果您正在使用：	本指南中与您相关的部分有：
• 任何SDK或工具	AWS SDK 和工具维护政策
• AWS Cloud9 • AWS CDK • AWS Toolkit for Azure DevOps • AWS Toolkit for JetBrains • AWS Toolkit for Visual Studio • AWS Toolkit for Visual Studio Code • AWS Serverless Application Model • AWS CodeArtifact • AWS CodeBuild • Amazon CodeCatalyst • AWS CodeCommit • AWS CodeDeploy	配置 身份验证和访问 AWS SDK 和工具维护政策

如果您正在使用：	本指南中与您相关的部分有：
• AWS CodePipeline	
• AWS CLI • AWS SDK for C++ • AWS SDK for Go • AWS SDK for Java • AWS SDK for JavaScript • AWS SDK for Kotlin • AWS SDK for .NET • AWS SDK for PHP • AWS SDK for Python (Boto3) • AWS SDK for Ruby • AWS SDK for Rust • AWS SDK for Swift • AWS Tools for Windows PowerShell	配置 身份验证和访问 设置参考 AWS 常用运行时 (CRT) 库 AWS SDK 和工具维护政策 AWS SDKs和工具版本支持

开发人员资源

有关可帮助您开发应用程序的工具的概述 AWS，请参阅构建工具 AWS。有关支持的信息，请参阅AWS 知识中心。

Amazon Q Developer 是一款基于人工智能的生成式对话助手，可以帮助您理解、构建、扩展和操作 AWS 应用程序。为了加快您的构建 AWS，为Amazon Q提供支持的模型增加了高质量的 AWS 内容，以生成更完整、更具可操作性和参考性的答案。有关更多信息，请参阅 Amazon Q 开发人员用户指南中的 什么是 Amazon Q 开发者版？。

工具包遥测通知

AWS 集成开发环境 (IDE) 工具包是允许访问您IDE中的 AWS 服务的插件和扩展。Amazon Q IDE 插件和扩展程序可在您的 AI 中提供生成式人工智能帮助IDE。有关每个IDE工具包的详细信息，请参阅上表中的 Toolkit 用户指南。要了解有关在中使用 Amazon Q 的更多信息IDE，请参阅 Amazon Q 开发者指南IDE主题中的使用 Amazon Q。

AWS IDEToolkits 和 Amazon Q 可能会收集和存储客户端遥测数据，以便为有关 future Toolkit AWS 和 Amazon Q 版本的决策提供依据。收集的数据可以量化您对 AWS 工具包和 Amazon Q 的使用情况。

要详细了解在所有 AWS IDE工具包和 Amazon Q 中收集的遥测数据，请参阅 Github 存储库中的 commonDefinitions.json 文档。`aws-toolkit-common`

有关每个 AWS IDE工具包和 Amazon Q 扩展程序收集的遥测数据的详细信息，请参阅以下 Too AWS I GitHub kit 存储库中的资源文档：

- AWS 带有 Amazon Q 的 Visual Stud
- AWS Toolkit for Visual Studio Code 还有适用于 VS Code 的 Amazon Q 扩展
- AWS Toolkit for JetBrains 还有适用于 Amazon Q 的插件 JetBrains
- 适用于 Eclipse 的 Amazon Q

AWS 工具包中可访问的某些 AWS 服务可能会收集额外的客户端遥测数据。有关每项 AWS 服务收集的数据类型的详细信息，请参阅您感兴趣的特定服务的AWS 文档主题。

配置

使用 AWS 软件开发工具包和其他 AWS 开发者工具（例如 AWS Command Line Interface (AWS CLI)），您可以与 AWS 服务 API 进行交互。但是，在尝试执行此操作之前，必须使用执行请求的操作所需的信息来配置 SDK 或工具。

这些信息包含以下各项：

- 识别 API 的调用方的凭证信息。凭据用于加密向 AWS 服务器发出的请求。使用此信息 AWS 确认您的身份，并可以检索与之相关的权限策略。然后，它可以确定允许您执行哪些操作。
- 其他配置详细信息，用于告知 AWS CLI 或 SDK 如何处理请求、将请求发送到何处（发送到哪个 AWS 服务端点）以及如何解释或显示响应。

每个 SDK 或工具都支持多个来源，您可以使用这些来源来提供所需的凭证和配置信息。有些来源是 SDK 或工具所独有的，您必须参阅该工具或 SDK 的文档，详细了解如何使用该方法。

但是，大多数 AWS SDK 和工具都支持来自两个主要来源（除了代码本身）的常见设置：

- 共享 AWS 配置和凭据文件-共享`credentials`文件`config`和文件是为 AWS SDK 或工具指定身份验证和配置的最常用方式。使用这些文件存储您的工具和应用程序可以使用的设置。共享文件 `config` 和 `credentials` 中的设置与特定的配置文件相关联。使用多个配置文件，您可以创建不同的设置配置以应用于不同的场景。当你使用 AWS 工具调用命令或使用 SDK 调用 AWS API 时，你可以指定用于该操作的配置文件以及配置设置。其中一个配置文件被指定为 `default` 配置文件，当您未明确指定要使用的配置文件时，将自动使用此配置文件。此参考指南中记录了您可以存储在这些文件中的设置。
- 环境变量 – 某些设置也可以存储在操作系统的环境变量中。尽管一次只能有一组有效的环境变量，但是随着程序的运行和需求的变化，可以很容易地对其进行动态修改。

此部分中的其他主题

- 共享config文件和credentials文件
- 共享文件 config 和 credentials 的位置
- 环境变量支持
- JVM 系统属性支持

共享config文件和credentials文件

共享credentials文件 AWS config和文件包含一组配置文件。配置文件是一组按键值对组成的配置设置，由 AWS Command Line Interface (AWS CLI) AWS SDKs、和其他工具使用。配置值附加到配置文件中，以便在使用 SDK /tool 配置文件时配置该配置文件的某些方面。这些文件是 “共享” 的，因为这些值对任何应用程序、进程或SDKs对用户的本地环境都有影响。

共享credentials文件config和文件都是纯文本文件，仅包含ASCII字符（UTF-8 编码）。它们采用通常所谓的INI文件的形式。

配置文件

共享文件 config 和 credentials 中的设置与特定的配置文件相关联。可以在文件中定义多个配置文件，以创建不同的设置配置以应用于不同的开发环境。

如果未指定特定的命名[default]配置文件，则该配置文件包含SDK或工具操作所使用的值。您也可以创建单独的配置文件，您可以按名称明确引用这些配置文件。每个配置文件都可以根据您的应用程序和场景的需要使用不同的设置和值。

> **Note**
>
> [default]只是一个未命名的配置文件。default之所以命名此配置文件，是因为SDK如果用户未指定配置文件，则它是使用的默认配置文件。它不为其他配置文件提供接替的默认值。如果您在[default]配置文件中设置了某些内容，但未在命名配置文件中进行设置，则在使用命名配置文件时不会设置该值。

设置已命名的个人资料

[default]配置文件和多个已命名的配置文件可以存在于同一个文件中。使用以下设置来选择您的SDK或工具在运行代码时使用哪些配置文件设置。配置文件也可以在代码中选择，或者在使用时按命令选择。 AWS CLI

通过设置以下选项之一来配置此功能：

AWS_PROFILE-环境变量

当此环境变量设置为命名配置文件或 “默认” 时，所有SDK代码和 AWS CLI 命令都使用该配置文件中的设置。

Linux/macOS 通过命令行设置环境变量的示例：

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

Windows 通过命令行设置环境变量的示例：

```
setx AWS_PROFILE "my_default_profile_name"
```

`aws.profile`-JVM 系统属性

SDK对于适用于 Kotlin JVM 和SDK适用于 Java 2.x 的，你可以设置`aws.profile`系统属性。SDK创建服务客户端时，它将使用命名配置文件中的设置，除非该设置在代码中被覆盖。SDK适用于Java 的 1.x 不支持此系统属性。

> **Note**
>
> 如果您的应用程序位于运行多个应用程序的服务器上，我们建议您始终使用命名配置文件而不是默认配置文件。环境中的任何 AWS 应用程序都会自动获取默认配置文件，并在它们之间共享。因此，如果其他人更新了其应用程序的默认配置文件，则可能会无意中影响其他应用程序。为了防范这种情况，请在共享`config`文件中定义一个命名的配置文件，然后通过在代码中设置命名配置文件来在应用程序中使用该命名配置文件。如果您知道命名的配置文件的作用域只会影响您的应用程序，则可以使用环境变量或JVM系统属性来设置该配置文件。

配置文件的格式

`config` 文件将归入各个节中。节是一个命名的设置集合，它一直持续到遇到另一个节定义行为止。

`config` 文件是使用以下格式的纯文本文件：

- 节中的所有条目均采用 `setting-name=value` 的一般形式。
- 可以通过以井号字符 (#) 开头来注释掉行。

节类型

节定义是将名称应用于设置集合的行。节定义行以方括号 ([]) 开头和结尾。方括号内有一个节类型标识符和该节的自定义名称。可以使用字母、数字、连字符 (-) 和下划线 (_)，但不能使用空格。

节类型：**default**

节定义行示例：[default]

[default]是唯一不需要分profile区标识符的配置文件。

下面的示例介绍一个有 [default] 配置文件的基本 config 文件。它设置了region设置。在此行之后的所有设置，直到遇到其他分区定义为止，均属于此配置文件的一部分。

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

节类型：**profile**

节定义行示例：[profile *dev*]

profile章节定义行是一个命名的配置分组，您可以将其应用于不同的开发场景。要更好地了解命名配置文件，请参阅前面关于配置文件的部分。

以下示例显示了一个config包含profile截面定义行和名为的命名截面的文件foo。在此行之后的所有设置，直到遇到另一个分区定义之前，都是此命名配置文件的一部分。

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

某些设置有自己的嵌套子设置组，例如以下示例中的 s3 设置和子设置。通过缩进一个或多个空格将子设置与组相关联。

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

节类型：**sso-session**

节定义行示例：[sso-session *my-sso*]

sso-session部分定义行命名了一组设置，您使用这些设置来配置配置文件以解析 AWS 凭据 AWS IAM Identity Center。有关配置单点登录身份验证的更多信息，请参阅 IAM您的SDK或工具的身份中心

身份验证。配置文件通过键值对链接到 sso-session 节，其中 sso-session 是密钥，您的 sso-session 节名称是值，例如 sso-session = <name-of-sso-session-section>。

以下示例配置了一个配置文件，该配置文件将使用 “m SampleRole y-sso” 中的令牌获取 “111122223333” 账户中 “” IAM 角色的短期 AWS 证书。在profile节中，使用 sso-session 密钥按名称引用 “my-sso” sso-session 节。

```
[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
```

节类型：**services**

节定义行示例：[services dev]

> **Note**
>
> 该services部分支持服务特定的端点自定义，并且仅在包含此功能的工具中SDKs可用。要查看此功能是否适用于您SDK，请参阅了解兼容 AWS SDKs服务特定的终端节点。

services部分定义行命名了一组为 AWS 服务 请求配置自定义终端节点的设置。配置文件通过键值对链接到 services 节，其中 services 是密钥，您的 services 节名称是值，例如 services = <name-of-services-section>。

该services部分进一步按<SERVICE> = 行分成小节，其中<SERVICE>是标 AWS 服务 识键。标 AWS 服务 识符基于API模型的标识符，将所有空格serviceId替换为下划线，并将所有字母小写。有关 services 节中要使用的所有服务标识符密钥的列表，请参阅特定于服务的端点的标识符。服务标识符密钥后面是嵌套的设置，每个设置单独成行，缩进两个空格。

以下示例使用services定义来配置端点，使其仅用于向 Amazon DynamoDB 服务发出的请求。在"local-dynamodb" services节中，使用services密钥按名称引用profile节。标 AWS 服务识符密钥是dynamodb。 Amazon DynamoDB 服务小节从线路开始dynamodb = 。后面紧跟的任何缩进行都包含在该小节中，并适用于该服务。

```
[profile dev]
```

```
services = local-dynamodb

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

有关自定义端点配置的更多信息，请参阅特定于服务的端点。

凭证文件的格式

credentials文件的规则通常与config文件的规则相同，唯一的不同是配置文件部分不是以单词profile开头。在方括号之间仅使用配置文件名称本身。以下示例显示了一个名为的命名配置credentials文件部分的文件foo。

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

只有以下被视为 “机密” 或敏感的设置才能存储在credentials文件中：aws_access_key_idaws_secret_access_key、和aws_session_token。尽管也可以将这些设置放在共享config文件中，但我们建议您将这些敏感值保存在单独credentials的文件中。这样，如有必要，您可以为每个文件提供单独的权限。

下面的示例介绍一个有 [default] 配置文件的基本 credentials 文件。它设置aws_access_key_idaws_secret_access_key、和aws_session_token全局设置。

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

无论您在文件中使用命名配置credentials文件还是 default “”，此处的所有设置都将与文件中使用相同配置config文件名称的任何设置合并。如果两个文件中都有共享相同名称的配置文件的凭证，则凭证文件中的密钥优先。

共享文件 config 和 credentials 的位置

共享credentials文件 AWS config和文件是纯文本文件，其中包含软件开发工具包和 AWS 工具的配置信息。这些文件位于您的本地环境中，由 SDK 代码或在该环境中运行的 AWS CLI 命令自动使用。例如，在您自己的计算机上或在 Amazon 弹性计算云实例上进行开发时。

当 SDK 或工具运行时，它会检查这些文件并加载所有可用的配置设置。如果这些文件尚不存在，则 SDK 或工具会自动创建基本文件。

默认情况下，这些文件位于名为的文件夹中.aws，该文件夹位于您的home或用户文件夹中。

操作系统	文件的默认位置和名称
Linux 和 macOS	~/.aws/config ~/.aws/credentials
Windows	%USERPROFILE%\.aws\config %USERPROFILE%\.aws\credentials

主目录解析

~仅在以下情况下才用于主目录解析：

- 开始路径
- 紧随其后的是/或平台特定的分隔符。在 Windows 上~/，~\两者都解析到主目录。

在确定主目录时，会检查以下变量：

- （所有平台）HOME 环境变量
- （Windows 平台）USERPROFILE 环境变量
- （Windows 平台）HOMEDRIVE和HOMEPATH环境变量的串联 () $HOMEDRIVE$HOMEPATH
- （可选，根据 SDK 或工具）特定于 SDK 或工具的主路径解析函数或变量

如有可能，如果在路径开头指定了用户的主目录（例如，~username/），则会将其解析到请求的用户名的起始目录（例如，/home/username/.aws/config）。

更改这些文件的默认位置

您可以使用以下任一方法来覆盖 SDK 或工具加载这些文件的位置。

使用环境变量

可以设置以下环境变量，将这些文件的位置或名称从默认值更改为自定义值：

- config 文件环境变量：**AWS_CONFIG_FILE**
- credentials 文件环境变量：**AWS_SHARED_CREDENTIALS_FILE**

Linux/macOS

您可以通过在 Linux 或 macOS 上运行以下导出命令来指定备用位置。

```
$ 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

您可以通过在 Windows 上运行以下setx命令来指定备用位置。

```
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
```

有关使用环境变量配置系统的更多信息，请参阅环境变量支持。

使用 JVM 系统属性

对于在 JVM 上运行的 Kotlin 开发工具包和适用于 Java 2.x 的 SDK，您可以设置以下 JVM 系统属性，将这些文件的位置或名称从默认值更改为自定义值：

- config文件 JVM 系统属性：**aws.configFile**
- credentials 文件环境变量：**aws.sharedCredentialsFile**

有关如何设置 JVM 系统属性的说明，请参阅the section called “如何设置 JVM 系统属性”。适用于 Java 的 SDK 1.x 不支持这些系统属性。

环境变量支持

环境变量提供了另一种指定配置选项和凭证的方法；若要编写脚本或将一个命名配置文件临时设置为默认配置文件，环境变量会很有用。有关大多数支持的环境变量的列表SDKs，请参阅环境变量列表。

选项的优先顺序

- 如果您使用环境变量来指定设置，则它将覆盖从共享配置文件加载的任何值 AWS `config`和`credentials`文件。
- 如果您通过使用上的参数来指定设置 AWS CLI 命令行，它会覆盖相应环境变量或配置文件中配置文件中的任何值。

如何设置环境变量

下面的示例介绍您如何可以为默认用户配置环境变量。

Linux, macOS, or Unix

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

设置环境变量会更改使用的值，直到 Shell 会话结束或直到您将该变量设置为其他值。通过在 shell 的启动脚本中设置变量，可使变量在未来的会话中继续有效。

Windows Command Prompt

```
C:\> setx AWS_ACCESS_KEY_ID AKIAIOSFODNN7EXAMPLE
C:\> setx AWS_SECRET_ACCESS_KEY wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
C:\> setx
 AWS_SESSION_TOKEN AQoEXAMPLEH4aoAH0gNCAPy...truncated...zrkuWJOgQs8IZZaIv2BXIa2R4Olgk
C:\> setx AWS_REGION us-west-2
```

使用 `set` 设置环境变量会更改使用的值，直到当前命令提示符会话结束，或者直到您将该变量设置为其他值。使用 `setx` 设置环境变量会更改当前命令提示符会话和运行该命令后创建的所有命令提示符会话中使用的值。它不 影响在运行该命令时已经运行的其他命令 shell。

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...zrkuWJOgQs8IZZaIv2BXIa2R40l
PS C:\> $Env:AWS_REGION="us-west-2"
```

如果您在 PowerShell 提示符处设置环境变量（如前面的示例所示），则它只会在当前会话的持续时间内保存该值。要使环境变量设置在所有会话 PowerShell 和命令提示符会话中保持不变，请使用控制面板中的系统应用程序将其存储。或者，您可以通过将变量添加到您的 PowerShell 个人资料中来为所有未来 PowerShell 会话设置该变量。有关存储环境变量或跨会话保存环境变量的更多信息，请参阅PowerShell 文档。

无服务器环境变量设置

如果您使用无服务器架构进行开发，则还有其他设置环境变量的选项。根据您的容器，您可以对在这些容器中运行的代码使用不同的策略来查看和访问环境变量，这与非云环境类似。

例如，用 AWS Lambda，你可以直接设置环境变量。有关详细信息，请参阅使用 AWS Lambda中的环境变量 AWS Lambda 开发者指南。

在 Serverless Framework 中，您通常可以在SDK环境设置下的提供者密钥下的`serverless.yml`文件中设置环境变量。有关该 `serverless.yml` 文件的信息，请参阅无服务器框架文档中的 常规功能设置 。

无论您使用哪种机制来设置容器环境变量，都有一些变量由容器保留，例如在 定义的运行时系统环境变量 中为 Lambda 记录的变量。请务必查阅您使用的容器的官方文档，以确定如何处理环境变量以及是否存在任何限制。

JVM 系统属性支持

JVM 系统属性提供了另一种为在 JVM 上运行的 SDK（例如和）指定配置选项和凭据的方法。 AWS SDK for Java AWS SDK for Kotlin有关 SDK 支持的 JVM 系统属性的列表，请参阅设置参考。

选项的优先顺序

- 如果您使用其 JVM 系统属性来指定设置，则该设置将覆盖在环境变量中找到或从共享 AWS `config`和`credentials`文件中的配置文件加载的任何值。

- 如果您使用环境变量来指定设置，则该设置将覆盖从共享 AWS `config` 和`credentials`文件中的配置文件加载的任何值。

如何设置 JVM 系统属性

您可以通过多种方式设置 JVM 系统属性。

在命令行上

使用开关调用命令时，在命令行上设置 JVM 系统属性。`java -D`除非您在代码中明确覆盖该值，否则以下命令将为所有服务客户端进行 AWS 区域 全局配置。

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

如果需要设置多个 JVM 系统属性，请多次指定`-D`交换机。

使用环境变量

如果您无法访问命令行来调用 JVM 来运行应用程序，则可以使用`JAVA_TOOL_OPTIONS`环境变量来配置命令行选项。这种方法在诸如在 Java 运行时上运行 AWS Lambda 函数或在嵌入式 JVM 中运行代码等情况下非常有用。

除非您在代码中明确覆盖该值，否则以下示例将为所有服务客户端进行 AWS 区域 全局配置。

Linux, macOS, or Unix

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

设置环境变量会更改使用的值，直到 Shell 会话结束或直到您将该变量设置为其他值。通过在 shell 的启动脚本中设置变量，可使变量在未来的会话中继续有效。

Windows Command Prompt

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

使用 `set` 设置环境变量会更改使用的值，直到当前命令提示符会话结束，或者直到您将该变量设置为其他值。使用 `setx` 设置环境变量会更改当前命令提示符会话和运行该命令后创建的所有命令提示符会话中使用的值。它不 影响在运行该命令时已经运行的其他命令 shell。

在运行时

您还可以在运行时使用以下示例所示的`System.setProperty`方法在代码中设置 JVM 系统属性。

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

Important

在初始化 SDK 服务客户端之前设置任何 JVM 系统属性，否则服务客户端可能会使用其他值。

身份验证和访问

使用开发 AWS 时，您必须确定您的代码是如何进行身份验证的。 AWS 服务您可以通过不同的方式配置对 AWS 资源的编程访问权限，具体取决于环境和可用的 AWS 访问权限。

本地（不在 AWS中）运行的代码的身份验证选项

- IAM您的SDK或工具的身份中心身份验证— 作为安全最佳实践，我们建议 AWS Organizations 与 Ident IAM ity Center 配合使用来管理所有人的访问权限 AWS 账户。你可以在中创建用户 AWS IAM Identity Center，使用 Microsoft Active Directory，使用 SAML 2.0 身份提供商 (IdP)，或者将你的 IdP 单独联合到其中。 AWS 账户要查看您的地区是否支持 Ident IAM ity Center，请参阅中的AWS IAM Identity Center 终端节点和配额Amazon Web Services 一般参考。
- IAM Roles Anywhere— 您可以使用 Ro IAM les Anywhere IAM 为在外部运行的服务器、容器和应用程序等工作负载获取临时安全证书 AWS。要在任何地方使用IAM角色，您的工作负载必须使用 X.509 证书。
- 扮演一个拥有 AWS 证书的角色— 您可以扮IAM演一个角色来临时访问原本可能无法访问的 AWS 资源。
- AWS 访问密钥— 其他可能不太方便或可能增加 AWS 资源安全风险的选项。

在 AWS 环境中运行的代码的身份验证选项

如果您的代码在上运行 AWS，则证书可以自动提供给您的应用程序。例如，如果您的应用程序托管在 Amazon Elastic Compute Cloud 上，并且存在与该资源关联的IAM角色，则证书将自动提供给您的应用程序。同样，如果您使用 Amazon ECS 或 Amazon EKS 容器，则可以通过容器内运行的代码通过凭证提供者链自动获取为该IAM角色设置SDK的证书。

- 为 Amazon EC2 实例使用IAM角色— 使用IAM角色在 Amazon EC2 实例上安全运行您的应用程序。
- 您可以通过以下方式 AWS 使用IAM身份中心进行编程交互：
  - AWS CloudShell用于从控制台运行 AWS CLI 命令。
  - 要尝试为软件开发团队提供基于云的协作空间，可以考虑使用 Amazon CodeCatalyst。

通过基于 Web 的身份提供者进行身份验证 - 移动或基于客户端的 Web 应用程序

如果您正在创建需要访问的移动应用程序或基于客户端的 Web 应用程序 AWS，请构建您的应用程序，使其能够使用 Web 联合身份验证动态请求临时 AWS 安全证书。

利用 Web 联合身份验证，您不需要创建自定义登录代码或管理自己的用户身份。相反，应用程序用户可以使用知名的外部身份提供商 (IdP) 登录，例如 Login with Amazon、Facebook、Google 或任何其他兼容 OpenID Connect () OIDC 的 IdP。他们可以接收身份验证令牌，然后将该令牌交换为 AWS 该映射中的临时安全证书，并分配给有权使用您的资源的IAM角色 AWS 账户。

要了解如何为您的SDK或工具进行配置，请参阅扮演具有网络身份或 OpenID Connect 的角色。

有关移动应用程序，请考虑使用 Amazon Cognito。Amazon Cognito 充当身份凭证代理程序并为您完成许多联合身份验证工作。有关更多信息，请参阅IAM用户指南中的将 Amazon Cognito 用于移动应用程序。

有关访问管理的更多信息

《IAM用户指南》包含以下有关安全控制 AWS 资源访问的信息：

- IAM身份（用户、用户组和角色）-了解中身份的基础知识 AWS。
- 中的安全最佳实践 IAM — 根据责任共担模型开发 AWS 应用程序时应遵循的安全建议。

Amazon Web Services 一般参考 具有以下基础知识：

- 了解并获取您的 AWS 证书-访问控制台和编程访问的密钥选项和管理实践。

AWS 构建者 ID

任何 AWS 账户 你可能已经拥有或想要创作的 AWS 构建者 ID 补充。虽然 AWS 账户 充当你创建的 AWS 资源的容器并为这些资源提供安全边界，但你的 AWS 构建者 ID 代表你是一个个体。您可以使用登录 AWS 构建者 ID 以访问开发者工具和服务，例如亚马逊 CodeWhisperer 和亚马逊 CodeCatalyst。

- 使用 AWS 构建者 IDAWS 登录 用户指南登录 — 了解如何创建和使用， AWS 构建者 ID 并了解生成器 ID 提供的内容。
- 使用 CodeWhisperer 和进行身份验证 AWS Toolkit -CodeWhisperer 用户指南中的生成器ID —了解如何 CodeWhisperer 使用. AWS 构建者 ID
- CodeCatalyst 概念- AWS 构建者 ID 在 Amazon CodeCatalyst 用户指南中-了解如何 CodeCatalyst 使用 AWS 构建者 ID.

IAM您的SDK或工具的身份中心身份验证

AWS IAM Identity Center 是在非AWS 计算服务上进行开发时推荐的提供 AWS 凭证的方法。例如，这将类似于您的本地开发环境。如果您正在使用诸如亚马逊弹性计算云 (AmazonEC2) 之类的 AWS 资源进行开发 AWS Cloud9，我们建议您改为从该服务获取证书。

在本教程中，您将建立 I IAM dentity Center 访问权限，并将使用 AWS 访问门户和为您的SDK工具配置该访问权限 AWS CLI。

- AWS 访问门户是您手动登录IAM身份中心的 Web 位置。的格式URL是`d-xxxxxxxxxx.awsapps.com/start`或*`your_subdomain`*`.awsapps.com/start`。登录 AWS 访问门户后，您可以查看 AWS 账户 已为该用户配置的角色。此过程使用 AWS 访问门户获取 SDK / tool 身份验证过程所需的配置值。
- 用于将您的SDK或工具配置为对您的代码发出的API呼叫使用 IAM Identity Center 身份验证。 AWS CLI 此一次性过程会更新您的共享 AWS `config`文件，然后在您运行代码时由您的SDK或工具使用该文件。

使用IAM身份中心配置编程访问权限

步骤 1：建立访问权限并选择相应的权限集

如果您尚未启用 IAM Identity Center，请参阅AWS IAM Identity Center 用户指南中的启用IAM身份中心。

选择以下方法之一来访问您的 AWS 证书。

我没有通过IAM身份中心建立访问权限

1. 按照用户指南中的使用默认 Identity C IAM enter 目录配置用户访问权限过程添加AWS IAM Identity Center 用户并添加管理权限。
2. `AdministratorAccess`权限集不应用于常规开发。相反，我们建议使用预定义的`PowerUserAccess`权限集，除非您的雇主为此目的创建了自定义权限集。

   再次按照使用默认 Identity C IAM enter 目录配置用户访问权限的步骤进行操作，但这一次：

   - 与其创建*`Admin team`*群组，不如创建一个*`Dev team`*群组，然后将其替换为说明中。
   - 您可以使用现有用户，但必须将该用户添加到新*`Dev team`*组中。

- 与其创建*AdministratorAccess*权限集，不如创建一个*PowerUserAccess*权限集，然后将其替换为说明中的权限集。

完成后，你应该有以下几点：

- 一个Dev team小组。
- Dev team群组的附加PowerUserAccess权限集。
- 您的用户已加入群Dev team组。

3. 退出门户并再次登录以查看您的 AWS 账户 和Administrator或选项PowerUserAccess。PowerUserAccess在使用您的工具时选择/ SDK。

我已经 AWS 可以通过雇主管理的联合身份提供商（例如 Microsoft Entra 或 Okta）进行访问

AWS 通过身份提供商的门户网站登录。如果您的云管理员已授予您PowerUserAccess（开发者）权限，则您 AWS 账户 会看到您有权访问的权限和权限集。在您的权限集名称旁边，可以看到有关使用该权限集手动或以编程方式访问账户的选项。

自定义实现可能会产生不同的体验，例如不同的权限集名称。如果您不确定要使用哪个权限集，请联系 IT 团队以寻求帮助。

我已经 AWS 可以通过雇主管理的 AWS 访问门户进行访问

AWS 通过 AWS 访问门户登录。如果您的云管理员已向您授予 PowerUserAccess（开发人员）权限，您将看到您有权访问的 AWS 账户 和您的权限集。在您的权限集名称旁边，可以看到有关使用该权限集手动或以编程方式访问账户的选项。

我已经 AWS 可以通过雇主管理的联合自定义身份提供商进行访问

请联系您的 IT 团队以寻求帮助。

步骤 2：配置SDKs和使用IAM身份中心的工具

1. 在您的开发计算机上安装最新的 AWS CLI。

   a. 参阅 AWS Command Line Interface 用户指南中的安装或更新最新版本的 AWS CLI。

   b. （可选）要验证是否 AWS CLI 正在运行，请打开命令提示符并运行该aws --version命令。

2. 登录 AWS 访问门户。您的雇主可能会提供此信息，URL或者您可以按照第 1 步：建立访问权限通过电子邮件获得。如果没有，请在的控制面板URL上找到您的AWS 访问门户https://console.aws.amazon.com/singlesignon/。

   a. 在 AWS 访问门户的账户选项卡中，选择要管理的个人账户。将显示您的用户的角色。选择访问密钥以获取命令行凭证或相应权限集的编程访问权限。使用预定义的 PowerUserAccess 权限集，或者您或您的雇主创建的任何权限集，以将最低权限应用于开发。

   b. 在获取凭证对话框中，选择 MacOS 和 Linux 或 Windows，具体取决于您的操作系统。

   c. 选择 IAMIdentity Center 凭证方法以获取下一步所需的`Issuer URL`和`SSO Region`值。注意：`SSO Start URL`可以与互换使用。`Issuer URL`

3. 在 AWS CLI 命令提示符下，运行`aws configure sso`命令。出现提示时，输入在上一步中收集的配置值。有关此 AWS CLI 命令的详细信息，请参阅使用`aws configure sso`向导配置您的个人资料。

   a. 对于提示`SSO Start URL`，请输入您获得的值`Issuer URL`。

   b. 对于CLI个人资料名称，我们建议您在开始*`default`*时输入。有关如何设置非默认（已命名）配置文件及其关联环境变量的信息，请参阅 配置文件。

4. （可选）在 AWS CLI 命令提示符下，通过运行`aws sts get-caller-identity`命令确认活动会话身份。响应应显示您配置的IAM身份中心权限集。

5. 如果您使用的是 AWS SDK，请在您的开发环境SDK中为您创建一个应用程序。

   a. 对于某些人来说SDKs，在使用 Identity Center IAM 身份验证之前，`SSOOIDC`必须将其他软件包（例如`SSO`和）添加到您的应用程序中。有关详细信息，请参阅您的具体内容SDK。

   b. 如果您之前配置了对的访问权限 AWS，请查看您的共享 AWS `credentials`文件是否有任何访问权限AWS 访问密钥。由于了解凭证提供商链优先顺序，在SDK或工具使用IAM身份中心凭证之前，您必须移除所有静态证书。

要深入了解SDKs和工具如何使用和使用此配置刷新凭据，请参阅了解 IAM Identity Center 身份验证。

根据您配置的会话时长，您的访问权限最终将过期，并且SDK或工具将遇到身份验证错误。要在需要时再次刷新访问门户会话， AWS CLI 请使用运行`aws sso login`命令。

您可以延长 Ident IAM ity Center 访问门户会话持续时间和权限集会话持续时间。这会延长您在需要再次使用 AWS CLI手动登录之前运行代码的时间。有关更多信息，请参阅《AWS IAM Identity Center 用户指南》中的以下主题：

- IAMIdentity Center 会话持续时间-配置用户 AWS 访问门户会话的持续时间

- 权限集会话持续时间 – 设置会话持续时间

有关所有 Ident IAM ity Center 提供商设置SDKs和工具的详细信息，请参阅本指南IAM身份中心凭证提供商中的。

了解 IAM Identity Center 身份验证

相关 IAM Identity Center 术语

以下术语可帮助您了解 AWS IAM Identity Center 背后的流程和配置。对于其中一些身份验证概念，AWS SDK API 的文档使用的名称与 IAM Identity Center 不同。知道这两个名字会很有帮助。

下表介绍了备用名称之间的关系。

IAM Identity Center 名称	SDK API 名称	描述
Identity Center	`sso`	尽管已重命名 AWS 单点登录，但出于向后兼容目的，`sso` API 命名空间仍将保留其原始名称。有关更多信息，请参阅 AWS IAM Identity Center 用户指南中的 IAM Identity Center 重命名。
IAM Identity Center 控制台 管理控制台		用于配置单点登录的控制台。
AWS 访问门户 URL		您的 IAM Identity Center 账户独有的 URL，例如 `https://xxx.awsapps.com/start` 。您使用您的 IAM Identity Center 登录凭证来登录此门户。
IAM Identity Center 访问门户会话	身份验证会话	向调用者提供持有者访问令牌。

IAM Identity Center 名称	SDK API 名称	描述
权限集会话		SDK 在内部用于进行 AWS 服务 调用的 IAM 会话。在非正式讨论中，您可能会看到它被错误地称为 “角色会话”。
权限集凭证	AWS 凭证 sigv4 凭证	SDK 实际用于大多数 AWS 服务 调用（特别是所有 sigv4 AWS 服务 调用）的凭证。在非正式讨论中，您可能会看到它被错误地称为 “角色凭证”。
IAM Identity Center 凭证提供者	SSO 凭证提供者	如何获取凭证，例如提供功能的类或模块。

了解 AWS 服务 的SDK 凭证解析

IAM Identity Center API 将持有者令牌凭证交换为 sigv4 凭证。大多数 AWS 服务 都是 sigv4 API，但也有一些例外，比如 Amazon CodeWhisperer 和 Amazon CodeCatalyst。以下内容描述了支持通过 AWS IAM Identity Center 对应用程序代码进行大多数 AWS 服务 调用的凭证解析过程。

开始 AWS 访问门户会话

- 使用您的凭证登录会话以开始该过程。
  - 使用 AWS Command Line Interface (AWS CLI) 中的 `aws sso login` 命令。如果您还没有活动会话，这将启动一个新的 IAM Identity Center 会话。
- 启动新会话时，您将收到来自 IAM Identity Center 的刷新令牌和访问令牌。AWS CLI 还会使用新的访问令牌和刷新令牌更新 SSO 缓存 JSON 文件，并使其可供 SDK 使用。
- 如果您已经有一个活动会话，则该 AWS CLI 命令将重复使用现有会话，且将在现有会话过期时过期。要了解如何设置 IAM Identity Center 会话的时长，请参阅AWS IAM Identity Center用户指南中的配置用户的 AWS 访问门户会话的持续时间。
  - 最大会话时长已延长至 90 天，以减少频繁登录的需求。

SDK 如何获取 AWS 服务 调用的凭证

当您为每个服务实例化客户端对象时，SDK 提供 AWS 服务 访问权限。将共享 AWS `config`文件的选定配置文件配置为 IAM Identity Center 凭证解析时，将使用 IAM Identity Center 来解析应用程序的凭证。

- 在创建客户端时，凭证解析过程将在运行时完成。

要使用 IAM Identity Center 单点登录检索 sigv4 API 的凭证，SDK 使用 IAM Identity Center 访问令牌获取 IAM 会话。此 IAM 会话称为权限集会话，它通过担任 IAM 角色提供对 SDK 的 AWS 访问权限。

- 权限集会话持续时间与 IAM Identity Center 会话持续时间是分开设置的。
  - 要了解如何设置权限集会话持续时间，请参阅AWS IAM Identity Center用户指南中的设置会话持续时间。
- 请注意，在大多数 AWS SDK API 文档中，权限集凭证也被称为AWS凭证和 sigv4 凭证。

权限集凭证通过调用 IAM Identity Center API 的 getRoleCredentials 返回到 SDK。SDK 的客户端对象使用该担任的 IAM 角色来调用 AWS 服务，例如让 Amazon S3 列出您账户中的桶。在权限集会话到期之前，客户端对象可以使用这些权限集凭证继续操作。

会话过期和刷新

使用 SSO令牌提供者配置 时，将使用刷新令牌自动刷新从 IAM Identity Center 获取的每小时访问令牌。

- 如果访问令牌在 SDK 尝试使用它时已过期，SDK 将使用刷新令牌来尝试获取新的访问令牌。IAM Identity Center 会将刷新令牌与您的 IAM Identity Center 访问门户会话持续时间进行比较。如果刷新令牌未过期，IAM Identity Center 将使用另一个访问令牌进行响应。
- 此访问令牌可用于刷新现有客户端的权限集会话，也可以用于解析新客户端的凭证。

但是，如果 IAM Identity Center 访问门户会话已过期，则不会授予新的访问令牌。因此，无法续订权限集持续时间。每当现有客户端的缓存权限集会话时长超时时，它就会过期（并且访问权限将丢失）。

在 IAM Identity Center 会话到期后，任何创建新客户端的代码都将无法通过身份验证。这是因为未缓存权限集凭证。在您拥有有效的访问令牌之前，您的代码将无法创建新客户端并完成凭证解析过程。

总而言之，当 SDK 需要新的权限集凭证时，SDK 会首先检查所有有效的现有凭证并使用这些凭证。无论凭证是针对新客户端，还是凭证已过期的现有客户端，这都适用。如果找不到凭证或凭证无效，则

SDK 会调用 IAM Identity Center API 来获取新凭证。要调用 API，它需要访问令牌。如果访问令牌已过期，SDK 会使用刷新令牌尝试从 IAM Identity Center 服务获取新的访问令牌。如果您的 IAM Identity Center 访问门户会话未过期，则会授予此令牌。

IAM Roles Anywhere

您可以使用 IAM Roles Anywhere 在 IAM 中获取临时安全凭证，以用于在 AWS 之外运行的服务器、容器和应用程序等工作负载。要使用 IAM Roles Anywhere，您的工作负载必须使用 X.509 证书。您的云管理员应提供所需的证书和私钥，以便将 IAM Roles Anywhere 配置为凭证提供者。

第 1 步：配置 IAM Roles Anywhere

IAM Roles Anywhere 提供了一种方法，用于获取在 AWS 外部运行的工作负载或流程的临时凭证。与证书颁发机构建立信任锚，以获取关联的 IAM 角色的临时凭证。该角色设置当您的代码使用 IAM Roles Anywhere 进行身份验证时您的工作负载将拥有的权限。

有关设置信任锚、IAM 角色和 IAM Roles Anywhere 配置文件的步骤，请参阅 IAM Roles Anywhere 用户指南中的在 AWS Identity and Access Management Roles Anywhere 中创建信任锚和配置文件。

> **Note**
>
> IAM Roles Anywhere 用户指南中的配置文件指的是 IAM Roles Anywhere 服务中的一个独特概念。它与共享的 AWS `config`文件中的配置文件无关。

第 2 步：使用 IAM Roles Anywhere

要从 IAM Roles Anywhere 获取临时安全凭证，请使用 IAM Roles Anywhere 提供的凭证助手。凭证工具可实现 IAM Roles Anywhere 的签名流程。

有关下载凭证助手工具的说明，请参阅 IAM Roles Anywhere 用户指南中的从 AWS Identity and Access Management Roles Anywhere 获取临时安全凭证。

要将从 IAM Roles Anywhere 获取的临时安全凭证与 AWS SDK 和 AWS CLI 一起使用，您可以在共享的 AWS `config`文件中配置`credential_process`设置。SDK 和 AWS CLI 支持使用 `credential_process` 进行身份验证的流程凭证提供者。下面显示了要设置`credential_process`的一般结构。

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

助手工具的 `credential-process` 命令以与 `credential_process` 设置兼容的标准 JSON 格式返回临时凭证。请注意，命令名称包含连字符，而设置名称包含下划线。命令需要使用以下参数：

- `private-key` – 签署请求的私钥的路径。
- `certificate` – 证书的路径。
- `role-arn` – 要为其获取临时凭证的角色的 ARN。
- `profile-arn` – 为指定角色提供映射的配置文件的 ARN。
- `trust-anchor-arn` – 用于身份验证的信任锚的 ARN。

您的云管理员将提供证书和私钥。所有三个 ARN 值都可以从 AWS Management Console 复制。以下示例显示了共享`config`文件，该文件配置了从助手工具检索临时凭证。

```
[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
```

有关可选参数和其他助手工具的详细信息，请参阅 GitHub 上的 IAM Roles Anywhere 凭证助手。

有关 SDK 配置设置本身和流程凭证提供者的详细信息，请参阅本指南中的 进程凭证提供者。

扮演一个拥有 AWS 证书的角色

假设角色涉及使用一组临时安全凭证来访问您原本无法访问的 AWS 资源。这些临时凭证由访问密钥 ID、秘密访问密钥和安全令牌组成。要了解有关 AWS Security Token Service (AWS STS) API 请求的更多信息，请参阅AWS Security Token Service API参考中的操作。

要将您的SDK或工具设置为代入角色，必须先创建或确定要担任的特定角色。IAM角色由角色唯一标识 Amazon 资源名称 (ARN)。角色与另一个实体建立信任关系。使用该角色的可信实体可能是一个 AWS 服务 或另一个实体 AWS 账户。要了解有关IAM角色的更多信息，请参阅IAM用户指南中的使用IAM角色。

确定IAM角色后，如果您受到该角色的信任，则可以将您的SDK或工具配置为使用该角色授予的权限。

> **Note**
>
> AWS 最佳做法是尽可能使用区域终端节点并配置您的终端节点AWS 区域。

IAM扮演一个角色

担任角色时， AWS STS 返回一组临时安全证书。这些凭证来自另一个配置文件或运行代码的实例或容器。最常见的是，当您拥有一个账户的 AWS 证书，但您的应用程序需要访问另一个账户中的资源时，会使用这种类型的角色代入方式。

步骤 1：设置IAM角色

要将您的SDK或工具设置为代入角色，必须先创建或确定要担任的特定角色。IAM角色是使用角色唯一标识ARN的。角色与另一个实体建立信任关系，通常是在您的账户内或用于跨账户访问。要进行设置，请参阅《IAM用户指南》中的创建IAM角色。

步骤 2：配置SDK或工具

将SDK或工具配置为从`credential_source`或获取凭据`source_profile`。

`credential_source`用于从亚马逊ECS容器、亚马逊EC2实例或环境变量中获取证书。

`source_profile` 用于从另一个配置文件获取凭证。`source_profile` 还支持角色链，即配置文件的层次结构，然后使用代入的角色来代入另一个角色。

当您在配置文件中指定此项时，SDK或工具会自动为您进行相应的 AWS STS `AssumeRole`API调用。要通过担任角色来检索和使用临时证书，请在共享 AWS `config`文件中指定以下配置值。有关这些设置各自的更多信息，请参阅 代入角色凭证提供者设置 节。

- `role_arn`-来自您在步骤 1 中创建的IAM角色
- 配置 `source_profile` 或 `credential_source`
- （可选）`duration_seconds`
- （可选）`external_id`
- （可选）`mfa_serial`

- （可选）role_session_name

以下示例显示了共享 config 文件中两个代入角色选项的配置：

```
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
```

有关所有代入角色凭证提供程序设置的详细信息，请参阅本指南中的 代入角色凭证提供者。

扮演具有网络身份或 OpenID Connect 的角色

假设角色涉及使用一组临时安全凭证来访问您原本无法访问的 AWS 资源。这些临时凭证由访问密钥 ID、秘密访问密钥和安全令牌组成。要了解有关 AWS Security Token Service (AWS STS) API 请求的更多信息，请参阅AWS Security Token Service API参考中的操作。

要将您的SDK或工具设置为代入角色，必须先创建或确定要担任的特定角色。 IAM角色由角色唯一标识 Amazon 资源名称 (ARN)。角色与另一个实体建立信任关系。使用该角色的可信实体可能是网络身份提供商、OpenID Connect (OIDC) 或SAML联合。要了解有关IAM角色的更多信息，请参阅IAM用户指南中的角色代入方法。

在中配置该IAM角色后SDK，如果将该角色配置为信任您的身份提供商，则可以进一步配置您的角色SDK以代入该角色以获得临时 AWS 证书。

> **Note**
>
> AWS 最佳做法是尽可能使用区域终端节点并配置您的终端节点AWS 区域。

使用 Web 身份或 OpenID Connect 进行联合

您可以使用公共身份提供商（例如 Login JSON With Amazon、Facebook、GoogleJWTs）提供的网络令牌 () 来获取临时 AWS 凭证AssumeRoleWithWebIdentity。根据它们的使用方式，它们JWTs可能被称为 ID 令牌或访问令牌。您也可以使用与发现协议兼容OIDC的身份提供商 (IdPs) JWTs 签发，例如 EntraId 或 PingFederate。

如果您使用的是 Amazon Elastic Kubernetes Service，则此功能允许您为亚马逊集群中的每个服务账户指定IAM不同的角色。EKS这个 Kubernetes 功能会分发JWTs到你的 pod 中，然后由该凭证提供者使用这些容器来获取临时证书。 AWS 有关此 Amazon EKS 配置的更多信息，请参阅《亚马逊EKS用户指南》中的服务账户IAM角色。但是，为了获得更简单的选择，如果您SDK支持，我们建议您改用 Amazon EKS Pod 身份。

步骤 1：设置身份提供商和IAM角色

要配置与外部 IdP 的联合，请使用IAM身份提供商 AWS 通知外部 IdP 及其配置。这将在您 AWS 账户和外部 IdP 之间建立信任。在配置SDK为使用 JSON Web 令牌 (JWT) 进行身份验证之前，必须先设置身份提供者 (IdP) 和用于访问它的IAM角色。要进行这些设置，请参阅IAM用户指南中的为网络身份或 OpenID Connect Federation（控制台）创建角色。

步骤 2：配置SDK或工具

将SDK或工具配置为使用来自的 JSON Web 令牌 (JWT) AWS STS 进行身份验证。

当您在配置文件中指定此项时，SDK或工具会自动为您进行相应的 AWS STS `AssumeRoleWithWebIdentity`API调用。要使用 Web 联合身份验证检索和使用临时证书，请在共享 AWS `config`文件中指定以下配置值。有关这些设置各自的更多信息，请参阅 代入角色凭证提供者设置 节。

- `role_arn`-来自您在步骤 1 中创建的IAM角色
- `web_identity_token_file` - 来自外部 IdP
- （可选）`duration_seconds`
- （可选）`role_session_name`

以下是使用 Web 身份代入角色的共享 `config` 文件配置示例：

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

> **Note**
>
> 有关移动应用程序，请考虑使用 Amazon Cognito。Amazon Cognito 充当身份凭证代理程序并为您完成许多联合身份验证工作。但是，Amazon Cognito 身份提供商不像其他身份提供商那样包含在SDKs和工具核心库中。要访问 Amazon CognitoAPI，请在你的或工具的版本或库中

加入亚马逊 Cognito 服务客户端。SDK有关与的用法 AWS SDKs，请参阅 Amazon Cognito 开发者指南中的代码示例。

有关所有代入角色凭证提供程序设置的详细信息，请参阅本指南中的 代入角色凭证提供者。

AWS 访问密钥

使用短期凭证

我们建议将您的 SDK 或工具配置为使用 IAM您的SDK或工具的身份中心身份验证 以使用延长的会话持续时间选项。

但是，要直接设置 SDK 或工具的临时凭证，请参阅 使用短期凭证进行身份验证。

使用长期凭证

> **Warning**
>
> 为了避免安全风险，在开发专用软件或处理真实数据时，请勿使用 IAM 用户进行身份验证，而是使用与身份提供商的联合身份验证，例如 AWS IAM Identity Center。

管理跨区域的访问权限 AWS 账户

作为安全最佳实践，我们建议 AWS Organizations 与 IAM Identity Center 配合使用来管理所有人的访问权限 AWS 账户。有关更多信息，请参阅《IAM 用户指南》中的 IAM 安全最佳实践。

你可以在 IAM Identity Center 中创建用户，使用 Microsoft Active Directory，使用 SAML 2.0 身份提供商 (IdP)，或者将你的 IdP 单独联合到其中。 AWS 账户您可以使用其中一种方法，为用户提供单点登录体验。您还可以强制执行多重身份验证 (MFA) 并使用临时证书 AWS 账户 进行访问。这与 IAM 用户不同，后者是一种可以共享的长期凭证，并且可能会增加 AWS 资源的安全风险。

仅为沙盒环境创建 IAM 用户

如果您不熟悉 AWS，可以创建一个测试 IAM 用户，然后使用它来运行教程并探索 AWS 所提供的内容。在学习时可以使用此类凭证，但我们建议您避免在沙盒环境之外使用。

对于以下用例，开始使用 IAM 用户可能是有意义的 AWS：

- 开始使用您的 AWS SDK 或工具，并在沙盒环境 AWS 服务 中进行探索。
- 在学习过程中，运行不支持人工参与登录流程的计划脚本、作业和其他自动化流程。

如果您在这些用例之外使用 IAM 用户，请尽快过渡到 IAM Identity Center 或将您的身份提供商联合到该 AWS 账户 中心。有关更多信息，请参阅 AWS中的身份联合验证。

确保 IAM 用户访问密钥安全

您应该定期轮换 IAM 用户访问密钥。参阅《IAM 用户指南》，按照轮换访问密钥中的指导进行操作。如果您认为自己不小心共享了您的 IAM 用户访问密钥，请轮换您的访问密钥。

IAM 用户访问密钥应存储在本地计算机上的共享 AWS `credentials`文件中。请勿将 IAM 用户访问密钥存储在您的代码中。请勿将包含 IAM 用户访问密钥的配置文件存储到任何源代码管理软件中。开源项目 git-secrets 等外部工具可以帮助您避免无意中将敏感信息提交到 Git 存储库。有关更多信息，请参阅 IAM 用户指南 中的 IAM 身份（用户、用户组和角色）。

要设置 IAM 用户以开始使用，请参阅 使用长期凭证进行身份验证。

使用短期凭证进行身份验证

我们建议将您的SDK或工具配置为IAM您的SDK或工具的身份中心身份验证与延长会话持续时间选项一起使用。但是，您可以复制和使用中提供的临时证书 AWS 访问门户。在这些临时凭证过期后，将需要复制新凭证。您可以在配置文件中使用临时凭证，也可以将其用作系统属性和环境变量的值。

最佳实践：我们建议您的应用程序使用从以下地址提供的临时证书，而不是手动管理凭证文件中的访问密钥和令牌：

- 网络 ACL 和安全组都允许 (因此可到达您的实例) 的发起 ping 的 AWS 计算服务，例如在 Amazon Elastic Compute Cloud 上或在 AWS Lambda.
- 证书提供商链中的另一种选择，例如。IAM您的SDK或工具的身份中心身份验证
- 或者使用进程凭证提供者来检索临时证书。

使用从中检索到的短期证书设置凭证文件 AWS 访问门户

1. 创建共享凭证文件。
2. 在凭证文件中，粘贴以下占位符文本，直到粘贴有效的临时凭证为止。

```
[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. 保存该文件。该 `~/.aws/credentials` 文件现在应该存在于您的本地开发系统上。如果未指定特定的命名配置文件，则此文件包含SDK或工具使用的 [默认] 配置文件。
4. 登录 AWS 访问门户。
5. 按照以下手动刷新凭据的说明从中复制IAM角色证书 AWS 访问门户。

   a. 对于链接说明中的第 4 步，请选择可根据您的开发需求授予访问权限的IAM角色名称。此角色的名称通常类似于PowerUserAccess或开发人员。

   b. 在链接说明中的第 7 步中，选择 “手动将个人资料添加到您的” AWS 凭证文件选项并复制内容。
6. 将复制的凭证粘贴到您的本地`credentials`文件中。如果您使用的是`default`配置文件，则不需要生成的配置文件名称。您的文件应类似于以下内容。

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

7. 保存 `credentials` 文件。

SDK创建服务客户端时，它将访问这些临时证书并将其用于每个请求。在步骤 5a 中选择的IAM角色的设置决定了临时证书的有效期限。最长持续时间为 12 小时。

在临时凭证过期后，重复步骤 4 到 7。

使用长期凭证进行身份验证

> **Warning**
>
> 为避免安全风险，在开发专用软件或处理真实数据时，请勿使用IAM用户进行身份验证。相反，请使用与身份提供商的联合，例如 AWS IAM Identity Center.

如果您使用IAM用户来运行代码，则开发环境中的SDK或工具将通过在共享环境中使用长期IAM用户证书进行身份验证 AWS `credentials`文件。查看IAM主题中的安全最佳实践，并尽快过渡到 IAM Identity Center 或其他临时证书。

有关凭证的重要警告和指南

有关凭证的警告

- 请务必NOT使用您账户的根凭证进行访问 AWS 资源的费用。这些凭证可提供不受限的账户访问且难以撤销。
- 请务必在应用程序文件中NOT输入字面访问密钥或凭据信息。如果您这样做，则在将项目上传到公共存储库或在其他情况下，会有意外暴露凭证的风险。
- 请NOT务必在项目区域中包含包含凭据的文件。
- 请注意，存储在共享中的所有凭据 AWS `credentials`文件以纯文本形式存储。

有关安全管理凭证的更多指南

有关如何安全管理的一般性讨论 AWS 凭证，请参阅管理的最佳实践 AWS中的访问密钥 AWS 一般参考。 除了讨论之外，还要考虑以下几点：

- 使用IAM角色执行亚马逊弹性容器服务 (AmazonECS) 任务的任务。
- 为IAM在 Amazon EC2 实例上运行的应用程序使用角色。

先决条件：创建 AWS account

使用IAM用户进行访问 AWS 服务，你需要一个 AWS 账户和 AWS 证书。

1. 创建账户。

   要创建 AWS 账户，参见入门：你是第一次吗 AWS 用户？ 在 AWS Account Management 参考指南。
2. 创建管理用户。

   请勿使用 root 用户账户（您创建的初始账户）访问管理控制台和服务。相反，请按照《用户指南》中的创建管理用户中所述创建一个管理IAM用户帐户。

   创建管理用户账户并记录登录详细信息后，务必注销根用户账户并使用管理账户重新登录。

这两个账户都不适合在上面进行开发 AWS 或者用于在上运行应用程序 AWS。 作为最佳实践，您需要创建适合这些任务的用户、权限集或服务角色。有关更多信息，请参阅《IAM用户指南》中的应用最低权限权限。

步骤 1：创建您的IAM用户

- 按照IAM用户指南中的创建IAM用户（控制台）过程创建IAM用户。创建IAM用户时：
  - 我们建议您选择 “向用户提供访问权限” AWS Management Console。 这允许您查看 AWS 服务 与你在可视环境中运行的代码有关，例如检查 AWS CloudTrail 诊断日志或将文件上传到 Amazon 简单存储服务，这在调试代码时很有用。
  - 在 “设置权限-权限选项” 中，选择 “直接附加策略”，以了解如何向该用户分配权限。
    - 大多数 “入门” SDK 教程都以 Amazon S3 服务为例。要向应用程序提供对 Amazon S3 的完全访问权限，请选择要附加到此用户的 `AmazonS3FullAccess` 策略。
  - 您可以忽略该过程中有关设置权限界限或标签的可选步骤。

步骤 2：获取您的访问密钥

1. 在IAM控制台的导航窗格中，选择用户，然后选择您之前创建**`User name`**的用户。
2. 在用户的页面上，选择安全凭证页面。然后，在访问密钥下，选择创建访问密钥。
3. 在创建访问密钥步骤 1 中，选择命令行界面 (CLI) 或本地代码。这两个选项生成的密钥类型相同，可与两个选项一起使用 AWS CLI 还有SDKs。
4. 对于创建访问密钥步骤 2，输入可选标记并选择下一步。
5. 在创建访问密钥步骤 3 中，选择下载.csv 文件以保存包含IAM用户访问密钥和私有访问密钥的`.csv`文件。稍后您将需要此信息。

   > **Warning**
   >
   > 使用适当的安全措施来确保这些凭证的安全。

6. 选择 Done (完成)。

步骤 3：更新共享 **`credentials`** 文件

1. 创建或打开共享 AWS `credentials`文件。此文件在 Linux 和 macOS 系统上为 `~/.aws/credentials`，在 Windows 上为 `%USERPROFILE%\.aws\credentials`。有关更多信息，请参阅 凭证文件位置。
2. 将以下文本添加到共享 `credentials` 文件中。将示例 ID 值和示例密钥值替换为先前下载的 `.csv` 文件中的值。

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

3. 保存该文件。

共享 `credentials` 文件是存储凭证的最常见方式。它们也可以设置为环境变量，有关环境变量的名称，请参阅 AWS 访问密钥。这是一种入门方法，但我们建议您尽快过渡到 Ident IAM ity Center 或其他临时证书。停止使用长期凭证后，请记得从共享 `credentials` 文件中删除这些凭证。

为 Amazon EC2 实例使用IAM角色

此示例介绍如何设置一个拥有 Amazon S3 访问权限的 AWS Identity and Access Management 角色，以便在部署到 Amazon EC2 实例的应用程序中使用。

要在亚马逊弹性计算云实例上运行您的 AWS SDK应用程序，请创建一个IAM角色，然后授予您的亚马逊EC2实例访问该角色的权限。有关更多信息，请参阅亚马逊EC2用户指南EC2中的亚马逊IAM角色。

创建 IAM 角色

您开发的 AWS SDK应用程序可能至少访问一个应用程序 AWS 服务 来执行操作。创建一个IAM角色来授予应用程序运行所需的权限。

例如，此过程创建了一个角色，该角色授予对 Amazon S3 的只读访问权限。许多 AWS SDK指南都有从 Amazon S3 中读出的 “入门” 教程。

1. 登录 AWS Management Console 并打开IAM控制台，网址为https://console.aws.amazon.com/iam/。
2. 在导航窗格中，选择角色，然后选择创建角色。
3. 对于 选择可信实体，在 可信实体类型，选择 AWS 服务。
4. 在 “用例” 下，选择 Amazon EC2，然后选择 “下一步”。
5. 对于添加权限，请从策略列表中选中 Amazon S3 只读访问权限复选框，然后选择下一步。
6. 输入角色的名称，然后选择创建角色。请记住这个名称，因为您在创建 Amazon EC2 实例时会用到它。

启动 Amazon EC2 实例并指定您的IAM角色

您可以通过执行以下操作使用您的IAM角色创建和启动 Amazon EC2 实例：

- 关注 Amazon EC2 用户指南中的快速启动实例。但是，在最后的提交步骤之前，还要执行以下操作：
  - 在高级详细信息下，在IAM实例配置文件中，选择您在上一步中创建的角色。

通过此设置IAM和 Amazon EC2 设置，您可以将应用程序部署到 Amazon EC2 实例，您的应用程序将拥有对 Amazon S3 服务的读取权限。

Connect 连接到EC2实例

连接到 Amazon EC2 实例，这样您就可以将应用程序传输到该实例，然后运行该应用程序。您需要包含您在创建实例时在 Key pair（登录）下使用的密钥对的私有部分的文件；也就是PEM文件。

为此，您可以按照实例类型的指导进行操作：连接到您的 Linux 实例或连接到您的 Windows 实例。当您连接时，请确保您可以将文件从开发计算机传输到您的实例。

> **Note**
>
> 在 Linux 或 macOS 终端上，您可以使用安全复制命令来复制您的应用程序。要scp与 key pair 一起使用，可以使用以下命令：`scp -i path/to/key file/to/copy ec2-user@ec2-xx-xx-xxx-xxx.compute.amazonaws.com:~`。
> 有关 Windows 的更多信息，请参阅将文件传输到 Windows 实例。

如果您使用的是 AWS 工具包，则通常也可以使用工具包连接到实例。有关更多信息，请参阅您使用的工具包的特定用户指南。

在EC2实例上运行您的应用程序

1. 将您的应用程序文件从本地驱动器复制到您的 Amazon EC2 实例。
2. 启动应用程序并验证其运行结果是否与开发计算机上的结果相同。
3. （可选）验证应用程序是否使用IAM角色提供的证书。
   a. 登录 AWS Management Console 并打开 Amazon EC2 控制台，网址为https://console.aws.amazon.com/ec2/。

b. 选择实例。

c. 选择 “操作”、“安全”，然后选择 “修改IAM角色”。

d. 对于IAM角色，请选择 “无IAM角色” 来分离IAM角色。

e. 选择更新IAM角色。

f. 再次运行该应用程序，并确认它返回了授权错误。

设置参考

SDKs提供特定于语言的内容APIs。 AWS 服务他们负责成功API拨打电话所需的一些繁重工作，包括身份验证、重试行为等。为此，SDKs他们需要制定灵活的策略来获取用于您的请求的凭证，维护用于每项服务的设置，以及获取用于全局设置的值。

您可以在以下各节中找到有关配置设置的详细信息：

- AWS SDKs和 Tools 标准化凭证提供商— 通用凭证提供商在多个SDKs证书提供者之间实现标准化。
- AWS SDKs和 “工具” 标准化功能— 通用功能跨多个标准化SDKs。

创建服务客户端

要以编程方式访问 AWS 服务，SDKs请为每个使用客户端类/对象。 AWS 服务例如，如果您的应用程序需要访问亚马逊EC2，则您的应用程序会创建一个 Amazon EC2 客户端对象来与该服务接口。然后，您可以使用服务客户端向该 AWS 服务发出请求。在大多数情况下SDKs，服务客户端对象是不可变的，因此您必须为向其发出请求的每个服务创建一个新的客户端，并使用不同的配置向同一服务发出请求。

设置的优先级

全局设置配置了大多数SDKs人支持并具有广泛 AWS 服务影响的功能、凭证提供程序和其他功能。所有地方SDKs都有一系列地点（或来源），他们会检查这些地点（或来源），以便找到全局设置的值。以下是设置查找优先级的方法：

1. 在代码中或服务客户端本身上设置的任何显式设置均优先于其他任何设置。
   - 有些设置可以根据每个操作进行设置，也可以根据需要针对调用的每个操作进行更改。对于 AWS CLI 或 AWS Tools for PowerShell，它们采用您在命令行上输入的每个操作参数的形式。对于 SDK，显式赋值可以采用您在实例化 AWS 服务 客户端或配置对象时或有时在调用个人时设置的参数的形式。API
2. 仅限 Java/Kotlin：已JVM选中该设置的系统属性。如果已设置该变量，将使用对应值配置客户端。
3. 系统会检查环境变量。如果已设置该变量，将使用对应值配置客户端。
4. SDK检查共享`credentials`文件中的设置。如果已设置，则客户端将使用它。
5. 设置的共享`config`文件。如果存在该设置，则SDK使用该设置。
   - `AWS_PROFILE`环境变量或`aws.profile`JVM系统属性可用于指定SDK加载哪个配置文件。

6. 最后使用SDK源代码本身提供的任何默认值。

> **Note**
>
> 有些SDKs工具可能会按不同的顺序进行检查。此外，有些SDKs和工具还支持其他存储和检索参数的方法。例如， AWS SDK for .NET 支持名为 SDKStor e 的额外来源。有关SDK或工具独有的提供商的更多信息，请参阅您正在使用的SDK或工具的特定指南。

顺序决定哪些方法优先使用并覆盖其他方法。例如，如果您在共享config文件中设置了配置文件，则只有在SDK或工具先检查其他位置后才能找到和使用该配置文件。这意味着，如果您在credentials文件中添加了设置，则会使用该设置而不是config文件中的设置。如果您使用设置和值配置环境变量，它将覆盖credentials和config文件中的该设置。最后，单个操作（AWS CLI命令行参数或API参数）或代码中的设置将覆盖该命令的所有其他值。

设置页面

本指南的 “设置” 参考部分中的页面详细介绍了可以通过各种机制进行设置的可用设置。下表列出了配置和凭证文件设置、环境变量以及（对于 Java 和 KotlinSDKs）可以在代码之外用于配置该功能的JVM设置。每个列表中的每个链接主题都会将您带到相应的设置页面。

- Config文件设置列表
- Credentials文件设置列表
- 环境变量列表
- JVM系统属性列表

每个凭证提供商或功能都有一个页面，其中列出了用于配置该功能的设置。对于每种设置，您通常可以通过将设置添加到配置文件或通过设置环境变量来设置值，或者（仅适用于 Java 和 Kotlin）通过设置JVM系统属性来设置值。每个设置都列出了所有支持的在描述详细信息上方的方块中设置该值的方法。尽管优先级各不相同，但无论如何设置，生成的功能都是一样的。

描述中将包括默认值（如果有），如果您什么都不做，该值就会生效。它还定义了该设置的有效值。

例如，让我们来看看请求压缩功能页面中的一个设置。

disable_request_compression示例设置的信息传达以下信息：

- 有三种等效的方法可以控制代码库之外的请求压缩。您可以：
  - 使用在您的配置文件中进行设置 `disable_request_compression`
  - 使用将其设置为环境变量 `AWS_DISABLE_REQUEST_COMPRESSION`
  - 或者，如果你使用的是 Java 或 KotlinSDK，请使用将其设置为JVM系统属性 `aws.disableRequestCompression`

> **Note**
>
> 可能还有一种方法可以直接在代码中配置相同的功能，但是本参考文献并未涵盖这一点，因为每种功能都是独一无二的SDK。如果要在代码本身中设置配置，请参阅您的特定SDK指南或API参考资料。

- 如果您什么都不做，则该值将默认为`false`。
- 此布尔值设置的唯一有效值是`true`和`false`。

每个功能页面的底部都有一个 “兼容性” AWS SDKs 表。

此表显示您是否SDK支持该页上列出的设置。该`Supported`列使用以下值表示支撑位：

- `Yes`— 所写的完全支持这些设置。SDK
- `Partial`— 支持某些设置或行为与描述有所不同。对于`Partial`，附加注释表示偏差。
- `No`— 不支持任何设置。这并不能说明代码中是否可以实现相同的功能；它仅表示不支持列出的外部配置设置。

Config文件设置列表

下表中列出的设置可以在共享 AWS `config`文件中分配。它们是全局性的，影响到所有 AWS 服务。SDKs而且工具还可能支持独特的设置和环境变量。要查看仅个人SDK或工具支持的设置和环境变量，请参阅该特定指南SDK或工具指南。

设置名称	详细信息	
`account_id_endpoint_mode`	基于账户的终端节点	

设置名称	详细信息
api_versions	常规配置设置
aws_access_key_id	AWS 访问密钥
aws_account_id	基于账户的终端节点
aws_secret_access_key	AWS 访问密钥
aws_session_token	AWS 访问密钥
ca_bundle	常规配置设置
credential_process	进程凭证提供者
credential_source	代入角色凭证提供者
defaults_mode	智能配置默认值
disable_request_compression	请求压缩
duration_seconds	代入角色凭证提供者
ec2_metadata_service_endpoint	IMDS凭证提供商
ec2_metadata_service_endpoint_mode	IMDS凭证提供商

设置名称	详细信息
ec2_metadata_v1_disabled	IMDS凭证提供商
endpoint_discovery_enabled	端点发现
endpoint_url	特定于服务的端点
external_id	代入角色凭证提供者
ignore_configured_endpoint_urls	特定于服务的端点
max_attempts	重试行为
metadata_service_num_attempts	Amazon EC2 实例元数据
metadata_service_timeout	Amazon EC2 实例元数据
mfa_serial	代入角色凭证提供者
output	常规配置设置
parameter_validation	常规配置设置
region	AWS 区域

设置名称	详细信息
request_min_compression_size_bytes	请求压缩
retry_mode	重试行为
role_arn	代入角色凭证提供者
role_session_name	代入角色凭证提供者
s3_disable_multiregion_access_points	Amazon S3 多区域访问点
s3_use_arn_region	Amazon S3 接入点
sdk_ua_app_id	应用程序 ID
source_profile	代入角色凭证提供者
sso_account_id	IAM身份中心凭证提供商
sso_region	IAM身份中心凭证提供商
sso_registration_scopes	IAM身份中心凭证提供商
sso_role_name	IAM身份中心凭证提供商
sso_start_url	IAM身份中心凭证提供商
sts_regional_endpoints	AWS STS 区域端点

设置名称	详细信息
use_dualstack_endpoint	双栈和端点 FIPS
use_fips_endpoint	双栈和端点 FIPS
web_identity_token_file	代入角色凭证提供者

Credentials文件设置列表

下表中列出的设置可以在共享 AWS credentials文件中分配。它们是全局性的，影响到所有 AWS 服务。SDKs而且工具还可能支持独特的设置和环境变量。要查看仅个人SDK或工具支持的设置和环境变量，请参阅该特定指南SDK或工具指南。

设置名称	详细信息
aws_access_key_id	AWS 访问密钥
aws_secret_access_key	AWS 访问密钥
aws_session_token	AWS 访问密钥

环境变量列表

下表列出了SDKs大多数人支持的环境变量。它们是全局性的，影响到所有 AWS 服务。SDKs而且工具还可能支持独特的设置和环境变量。要查看仅个人SDK或工具支持的设置和环境变量，请参阅该特定指南SDK或工具指南。

设置名称	详细信息
AWS_ACCESS_KEY_ID	AWS 访问密钥
AWS_ACCOUNT_ID	基于账户的终端节点
AWS_ACCOUNT_ID_ENDPOINT_MODE	基于账户的终端节点
AWS_CA_BUNDLE	常规配置设置
AWS_CONFIG_FILE	共享文件 config 和 credentials 的位置
AWS_CONTAINER_AUTHORIZATION_TOKEN	容器凭证提供者
AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE	容器凭证提供者
AWS_CONTAINER_CREDENTIALS_FULL_URI	容器凭证提供者
AWS_CONTAINER_CREDENTIALS_RELATIVE_URI	容器凭证提供者
AWS_DEFAULTS_MODE	智能配置默认值

设置名称	详细信息	
AWS_DISABLE_REQUEST_COMPRESSION	请求压缩	
AWS_EC2_METADATA_DISABLED	IMDS凭证提供商	
AWS_EC2_METADATA_SERVICE_ENDPOINT	IMDS凭证提供商	
AWS_EC2_METADATA_SERVICE_ENDPOINT_MODE	IMDS凭证提供商	
AWS_EC2_METADATA_V1_DISABLED	IMDS凭证提供商	
AWS_ENABLE_ENDPOINT_DISCOVERY	端点发现	
AWS_ENDPOINT_URL	特定于服务的端点	
AWS_ENDPOINT_URL_<SERVICE>	特定于服务的端点	

设置名称	详细信息
AWS_IGNORE_CONFIGURED_ENDPOINT_URLS	特定于服务的端点
AWS_MAX_ATTEMPTS	重试行为
AWS_METADATA_SERVICE_NUM_ATTEMPTS	Amazon EC2 实例元数据
AWS_METADATA_SERVICE_TIMEOUT	Amazon EC2 实例元数据
AWS_PROFILE	共享config和credentials 文件
AWS_REGION	AWS 区域
AWS_REQUEST_MIN_COMPRESSION_SIZE_BYTES	请求压缩
AWS_RETRY_MODE	重试行为
AWS_ROLE_ARN	代入角色凭证提供者
AWS_ROLE_SESSION_NAME	代入角色凭证提供者
AWS_S3_DISABLE_MULTIREGION_ACCESS_POINTS	Amazon S3 多区域访问点

设置名称	详细信息	
AWS_S3_USE_ARN_REGION	Amazon S3 接入点	
AWS_SDK_UA_APP_ID	应用程序 ID	
AWS_SECRET_ACCESS_KEY	AWS 访问密钥	
AWS_SESSION_TOKEN	AWS 访问密钥	
AWS_SHARED_CREDENTIALS_FILE	共享文件 config 和 credentials 的位置	
AWS_STS_REGIONAL_ENDPOINTS	AWS STS 区域端点	
AWS_USE_DUALSTACK_ENDPOINT	双栈和端点 FIPS	
AWS_USE_FIPS_ENDPOINT	双栈和端点 FIPS	
AWS_WEB_IDENTITY_TOKEN_FILE	代入角色凭证提供者	

JVM系统属性列表

您可以将以下JVM系统属性用于 AWS SDK for Java 和 AWS SDK for Kotlin （以JVM）为目标。有关如何设置JVM系统属性的说明，请参阅the section called “如何设置 JVM 系统属性”。

设置名称	详细信息
aws.accessKeyId	AWS 访问密钥
aws.accountId	基于账户的终端节点
aws.accountIdEndpointMode	基于账户的终端节点
aws.configFile	共享文件 config 和 credentials 的位置
aws.defaultsMode	智能配置默认值
aws.disableEc2MetadataV1	IMDS凭证提供商
aws.disableRequestCompression	请求压缩
aws.ec2MetadataServiceEndpoint	IMDS凭证提供商
aws.ec2MetadataServiceEndpointMode	IMDS凭证提供商
aws.endpointDiscoveryEnabled	端点发现
aws.endpointUrl	特定于服务的端点

设置名称	详细信息
aws.endpointUrl<ServiceName>	特定于服务的端点
aws.ignoreConfiguredEndpointUrls	特定于服务的端点
aws.maxAttempts	重试行为
aws.profile	共享config和credentials 文件
aws.region	AWS 区域
aws.requestMinCompressionSizeBytes	请求压缩
aws.retryMode	重试行为
aws.roleArn	代入角色凭证提供者
aws.roleSessionName	代入角色凭证提供者
aws.s3DisableMultiRegionAccessPoints	Amazon S3 多区域访问点
aws.s3UseArnRegion	Amazon S3 接入点
aws.secretAccessKey	AWS 访问密钥

设置名称	详细信息
aws.sessionToken	AWS 访问密钥
aws.sharedCredentialsFile	共享文件 config 和 credentials 的位置
aws.useDualstackEndpoint	双栈和端点 FIPS
aws.useFipsEndpoint	双栈和端点 FIPS
aws.userAgentAppId	应用程序 ID
aws.webIdentityTokenFile	代入角色凭证提供者

AWS SDKs和 Tools 标准化凭证提供商

许多凭证提供商已标准化为一致的默认值，并且在许多SDKs证书提供商中都以相同的方式工作。这种一致性可以提高跨多个编码时的生产力和清晰度SDKs。所有设置都可以在代码中被覆盖。有关详细信息，请参阅您的具体内容SDKAPI。

> **Important**
>
> 并非所有提供商都SDKs支持所有提供商，甚至支持提供商内部的所有方面。

主题

- 了解凭证提供商链
- SDK特定和特定工具的凭证提供者链

- AWS 访问密钥
- 代入角色凭证提供者
- 容器凭证提供者
- IAM身份中心凭证提供商
- IMDS凭证提供商
- 进程凭证提供者

了解凭证提供商链

所有SDKs人都有一系列地点（或来源）供他们检查，以便找到用于向某人提出请求的有效凭证 AWS 服务。找到有效凭证后，搜索即告停止。这种系统搜索被称为凭证提供者链。

使用其中一个标准化凭证提供商时， AWS SDKs始终尝试在证书到期时自动续订证书。无论您在链中使用哪个提供商，内置的凭证提供商链都使您的应用程序能够刷新您的凭证。无需其他代码即可执行SDK此操作。

尽管它们使用的不同链条各SDK不相同，但它们通常包括以下来源：

凭证提供者	描述
AWS 访问密钥	AWS IAM用户的访问密钥（例如`AWS_ACCESS_KEY_ID` 、和`AWS_SECRET_ACCESS_KEY` ）。
使用 Web 身份或 OpenID Connect 进行联合 - 承担角色凭证提供者	使用知名的外部身份提供商 (IdP) 登录，例如 Login with Amazon、Facebook、Google 或任何其他兼容 OpenID Connect () OIDC 的 IdP。使用 () 中的 JSON Web 令牌 (JWT) 假设IAM角色的 AWS Security Token Service 权限。AWS STS
IAM身份中心凭证提供商	从中获取凭证 AWS IAM Identity Center。
代入角色凭证提供者	通过承担IAM角色的权限来访问其他资源。（检索角色的临时凭证，然后使用该凭证）。
容器凭证提供者	亚马逊弹性容器服务（亚马逊ECS）和亚马逊 Elastic Kubernetes Service（亚马逊）凭证。EKS容器凭证提供者获取客户的容器化应用程序的凭证。

凭证提供者	描述
进程凭证提供者	自定义凭证提供者。从外部来源或进程（包括 IAM Roles Anywhere）获取凭证。
IMDS凭证提供商	亚马逊弹性计算云 (AmazonEC2) 实例配置文件凭证。将IAM角色与您的每个EC2实例相关联。在该实例上运行的代码就可以使用该角色的临时凭证。凭证通过 Amazon EC2 元数据服务提供。

对于链中的每个步骤，都有多种分配设置值的方法。在代码中指定的设置值始终优先。但是，还有 环境变量 和 共享config文件和credentials文件。有关更多信息，请参阅 设置的优先级。

SDK特定和特定工具的凭证提供者链

要直接访问您或工具SDK的特定凭证提供商链详细信息，请从以下选项中选择您的SDK或工具：

- AWS CLI
- SDK对于 C++
- SDKfor Go
- SDK适用于 Java
- SDK对于 JavaScript
- SDK对于 Kotlin 来说
- SDK对于。 NET
- SDK for PHP
- SDK适用于 Python (Boto3)
- SDK对于 Ruby
- SDK对于 Rust
- SDK为斯威夫特
- 用于 PowerShell

AWS 访问密钥

> **Warning**
>
> 为避免安全风险，在开发专用软件或处理真实数据时，请勿使用IAM用户进行身份验证。相反，请使用与身份提供商的联合，例如 AWS IAM Identity Center.

AWS IAM用户的访问密钥可以用作您的 AWS 证书。这些区域有： AWS SDK自动使用这些 AWS 用于签署API请求的凭据 AWS，这样您的工作负载就可以访问您的 AWS 安全便捷地提供资源和数据。建议始终使用`aws_session_token`，这样凭证才是临时的，过期后不再有效。不建议使用长期证书。

> **Note**
>
> 如果 AWS 无法刷新这些临时证书， AWS 可以延长凭证的有效期，这样您的工作负载就不会受到影响。

共享的 AWS `credentials`文件是存储凭据信息的推荐位置，因为它安全地位于应用程序源目录之外，并且与共享`config`文件的SDK特定设置是分开的。

了解相关更多信息 AWS 凭证和使用访问密钥，请参阅 AWS 《用户指南》中的安全证书和管理IAM用户的访问密钥。IAM

使用以下方法配置此功能：

`aws_access_key_id`-共享 AWS **`config`**文件设置, **`aws_access_key_id`**-共享 AWS **`credentials`**文件设置（推荐方法）, **`AWS_ACCESS_KEY_ID`** - 环境变量, **`aws.accessKeyId`**-JVM 系统属性：仅限 Java/Kotlin

指定 AWS 访问密钥用作证书的一部分，用于对用户进行身份验证。

`aws_secret_access_key`-共享 AWS **`config`**文件设置, **`aws_secret_access_key`**-共享 AWS **`credentials`**文件设置（推荐方法）, **`AWS_SECRET_ACCESS_KEY`** - 环境变量, **`aws.secretAccessKey`**-JVM 系统属性：仅限 Java/Kotlin

指定 AWS 作为证书一部分用于对用户进行身份验证的密钥。

aws_session_token-共享 AWS **config**文件设置, **aws_session_token**-共享 AWS **credentials**文件设置（推荐方法）, **AWS_SESSION_TOKEN** - 环境变量, **aws.sessionToken**-JVM 系统属性：仅限 Java/Kotlin

> 指定一个 AWS 会话令牌用作证书的一部分，用于对用户进行身份验证。您会收到此值作为成功请求承担角色所返回的临时凭证的一部分。只有在手动指定临时安全凭证时才需要会话令牌。但是，我们建议您始终使用临时安全凭证代替长期凭证。有关安全建议，请参阅中的安全最佳实践IAM。

有关如何获取这些值的说明，请参阅 使用短期凭证进行身份验证。

在config或credentials文件中设置这些必需值的示例：

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

Linux/macOS 通过命令行设置环境变量的示例：

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

Windows 通过命令行设置环境变量的示例：

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

与之兼容 AWS SDKs

以下内容SDKs支持本主题中描述的功能和设置。所有部分例外情况均已注明。支持任何JVM系统属性设置 AWS SDK for Java 还有 AWS SDK for Kotlin 只有。

SDK	支持	备注或更多信息
AWS CLI v2	是	
SDK对于 C++	是	不支持共享的config文件。
SDK适用于 Go V2 (1.x)	是	
SDK适用于 Go 1.x (V1)	是	要使用共享 config 文件设置，必须开启从配置文件加载的功能；请参阅会话。
SDK适用于 Java 2.x	是	
SDK适用于 Java 1.x	是	
SDK适用于 JavaScript 3.x	是	
SDK适用于 JavaScript 2.x	是	
SDK对于 Kotlin 来说	是	
SDK对于。 NET3.x	是	不支持环境变量。
SDK适用于 PHP 3.x	是	
SDK适用于 Python (Boto3)	是	
SDK适用于 Ruby 3.x	是	
SDK对于 Rust	是	
SDK为斯威夫特	是	
用于 PowerShell	是	不支持环境变量。

代入角色凭证提供者

假设角色涉及使用一组临时安全凭证来访问您原本无法访问的 AWS 资源。这些临时凭证由访问密钥 ID、秘密访问密钥和安全令牌组成。

要将您的SDK或工具设置为代入角色，必须先创建或确定要担任的特定角色。 IAM角色由角色唯一标识 Amazon 资源名称 (ARN)。角色与另一个实体建立信任关系。使用该角色的可信实体可能是 AWS 服务、另一个 AWS 账户、Web 身份提供商或OIDCSAML联合体。

确定IAM角色后，如果您受到该角色的信任，则可以将您的SDK或工具配置为使用该角色授予的权限。要执行此操作，请使用以下设置。

有关开始使用这些设置的指导，请参阅本指南中的 扮演一个拥有 AWS 证书的角色。

代入角色凭证提供者设置

使用以下方法配置此功能：

`credential_source`-共享 AWS **`config`**文件设置

在 Amazon EC2 实例或 Amazon 弹性容器服务容器中使用，用于指定SDK或工具可以在何处找到有权担任您使用`role_arn`参数指定的角色的证书。

默认值： 无

有效值：

- 环境-指定SDK或工具用于从环境变量`AWS_ACCESS_KEY_ID`和中检索源凭证`AWS_SECRET_ACCESS_KEY`。
- Ec@@ 2 InstanceMetadata — 指定SDK或工具将使用附加到EC2实例配置文件的IAM角色来获取源证书。
- EcsContainer— 指定SDK或工具将使用附加到ECS容器的IAM角色来获取源证书。

不能在同一配置文件中同时指定 `credential_source` 和 `source_profile`。

在`config`文件中设置此项以表明证书应来自亚马逊的示例EC2：

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

`duration_seconds`-共享 AWS **`config`**文件设置

指定角色会话的最大持续时间（以秒为单位）。

仅当配置文件指定代入角色时，此设置才适用。

默认值： 3600 秒 (1 小时)

有效值：该值的范围在 900 秒（15 分钟）到角色配置的最大会话持续时间（43200 秒或 12 小时）之间。有关更多信息，请参阅IAM用户指南中的查看角色的最大会话持续时间设置。

在 config 文件中设置此项的示例：

```
duration_seconds = 43200
```

external_id-共享 AWS **config**文件设置

指定第三方用于在其客户账户中代入角色的唯一标识符。

仅当配置文件指定代入角色且该角色的信任策略需要 ExternalId 值时，此设置才适用。该值映射到配置文件指定角色时传递给 AssumeRole 操作的 ExternalId 参数。

默认值：无。

有效值：请参阅IAM用户指南中的在向第三方授予对您的 AWS 资源的访问权限时如何使用外部 ID。

在 config 文件中设置此项的示例：

```
external_id = unique_value_assigned_by_3rd_party
```

mfa_serial-共享 AWS **config**文件设置

指定用户在担任角色时必须使用的多因素身份验证 (MFA) 设备的标识号或序列号。

在担任角色时，如果该角色的信任策略包含需要MFA身份验证的条件，则为必填项。有关的更多信息MFA，请参阅《IAM用户指南》IAM中的AWS 多重身份验证。

默认值：无。

有效值：该值可以是硬件设备的序列号（例如GAHT12345678），也可以是虚拟MFA设备的 Amazon 资源名称 (ARN)。的格式ARN是：arn:aws:iam::*account-id*:mfa/*mfa-device-name*

在 config 文件中设置此项的示例：

此示例假设一个名MyMFADevice为的虚拟MFA设备已为该账户创建并已为用户启用。

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

role_arn-共享 AWS **config**文件设置, **AWS_ROLE_ARN** - 环境变量, **aws.roleArn**-JVM 系统属性：仅限 Java/Kotlin

指定要用于执行使用此配置文件请求的操作的IAM角色的 Amazon 资源名称 (ARN)。

默认值：无。

有效值：该值必须是IAM角色ARN的值，格式如下：arn:aws:iam::*account-id*:role/*role-name*

此外，您还必须指定以下设置之一：

- source_profile - 标识另一个配置文件，用于查找具有在此配置文件中代入该角色的权限的凭证。
- credential_source— 使用由当前环境变量标识的凭证或附加到亚马逊EC2实例配置文件的凭证，或者使用亚马逊ECS容器实例。
- web_identity_token_file— 为已在移动或网络应用程序中进行身份验证的用户使用公共身份提供商或任何与 OpenID Connect (OIDC) 兼容的身份提供商。

role_session_name-共享 AWS **config**文件设置, **AWS_ROLE_SESSION_NAME** - 环境变量, **aws.roleSessionName**-JVM 系统属性：仅限 Java/Kotlin

指定要附加到角色会话的名称。此名称显示在与此会话关联的条目的 AWS CloudTrail 日志中，该会话可能在审核时有用。有关详细信息，请参阅《AWS CloudTrail 用户指南》中的CloudTrail userIdentity 元素。

默认值：可选参数。如果未提供此值，只要配置文件代入角色，则将自动生成会话名称。

有效值：在或代表您 AWS API调用AssumeRole操作（ AWS CLI 或诸如操作之类的AssumeRoleWithWebIdentity操作）时提供给RoleSessionName参数。该值成为您可以查询的代入角色用户 Amazon Resource Name (ARN) 的一部分，并作为该配置文件调用的操作的 CloudTrail 日志条目的一部分显示。

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

在 config 文件中设置此项的示例：

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

source_profile-共享 AWS **config**文件设置

指定其他配置文件，其凭证用于代入由原始配置文件中的 role_arn 设置指定的角色。要了解如何在共享credentials文件 AWS config和文件中使用配置文件，请参阅共享config文件和credentials文件。

如果您指定的配置文件也是代入角色配置文件，则将按顺序代入每个角色以完全解析凭证。当SDK遇到带有凭据的个人资料时，该链就会停止。角色链接将您的 AWS CLI 或 AWS API角色会话限制为最长一小时，并且无法延长。有关更多信息，请参阅IAM用户指南中的角色术语和概念。

默认值：无。

有效值：由 config 和 credentials 文件中定义的配置文件的名称组成的文本字符串。还必须在当前配置文件中指定 role_arn 的值。

不能在同一配置文件中同时指定 credential_source 和 source_profile。

在配置文件中设置此项的示例：

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

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

在前面的示例中，A配置文件告诉SDK或工具自动查找关联B配置文件的凭证。在这种情况下，B配置文件使用提供的凭证帮助工具IAM Roles Anywhere来获取凭证。 AWS SDK然后，代码会使用这些临时凭证来访问 AWS 资源。指定的角色必须附加允许请求的代码运行的IAM权限策略，例如命令 AWS 服务、或API方法。配置文件执行的每项操作的 CloudTrail 日志中A都包含角色会话名称。

关于角色链接的第二个示例，如果您在 Amazon Elastic Compute Cloud 实例上有一个应用程序，并且您想让该应用程序担任另一个角色，则可以使用以下配置。

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

```
role_session_name = ProfileARoleSession

[profile B]
credential_source=Ec2InstanceMetadata
```

Profile A 将使用来自 Amazon EC2 实例的证书担任指定角色，并将自动续订证书。

web_identity_token_file-共享 AWS **config**文件设置, **AWS_WEB_IDENTITY_TOKEN_FILE** - 环境变量, **aws.webIdentityTokenFile**-JVM 系统属性：仅限 Java/Kotlin

指定文件路径，该文件包含来自支持的 OAuth 2.0 提供商或 OpenID Connect ID 身份提供商的访问令牌。

此设置允许使用 Web 身份联合验证提供者（例如 Google、Facebook 和 Amazon 等）进行身份验证。SDK或开发者工具加载此文件的内容，并在代表您调用AssumeRoleWithWebIdentity操作时将其作为WebIdentityToken参数传递。

默认值：无。

有效值：此值必须是路径和文件名。该文件必须包含身份提供商向您提供的 OAuth 2.0 访问令牌或 OpenID Connect 令牌。相对路径被视为相对于进程工作目录的相对路径。

与之兼容 AWS SDKs

以下内容SDKs支持本主题中描述的功能和设置。所有部分例外情况均已注明。 AWS SDK for Java 和 AWS SDK for Kotlin 唯一支持任何JVM系统属性设置。

SDK	支持	备注或更多信息
AWS CLI v2	是	
SDK对于 C++	部分	credential_source 不支持。duration_seconds 不支持。mfa_serial 不支持。
SDK适用于 Go V2 (1.x)	是	
SDK适用于 Go 1.x (V1)	是	要使用共享 config 文件设置，必须开启从配置文件加载的功能；请参阅会话。

SDK	支持	备注或更多信息
SDK适用于 Java 2.x	部分	`mfa_serial` 不支持。 `duration_seconds` 不支持。
SDK适用于 Java 1.x	部分	`credential_source` 不支持。 `mfa_serial` 不支持。 JVM不支持系统属性。
SDK适用于 JavaScript 3.x	是	
SDK适用于 JavaScript 2.x	部分	`credential_source` 不支持。
SDK对于 Kotlin 来说	是	
SDK对于。 NET3.x	是	
SDK适用于 PHP 3.x	是	
SDK适用于 Python (Boto3)	是	
SDK适用于 Ruby 3.x	是	
SDK对于 Rust	是	
SDK为斯威夫特	是	
用于 PowerShell	是	

容器凭证提供者

容器凭证提供者为客户的容器化应用程序获取凭证。该证书提供者对亚马逊弹性容器服务（亚马逊 ECS）和亚马逊 Elastic Kubernetes Service（亚马逊）客户非常有用。EKSSDKs尝试通过GET请求从指定HTTP端点加载凭证。

如果您使用 AmazonECS，我们建议您使用任务IAM角色来改善凭证隔离、授权和可审计性。配置后，Amazon 会ECS设置SDKs和工具用来获取凭证的`AWS_CONTAINER_CREDENTIALS_RELATIVE_URI`环境变量。要ECS为亚马逊配置此功能，请参阅《亚马逊弹性容器服务开发者指南》中的任务IAM角色。

如果您使用 AmazonEKS，我们建议您使用 Amazon EKS Pod Identity 来改善凭证隔离、最低权限、可审计性、独立操作、可重用性和可扩展性。你的 Pod 和IAM角色都与 Kubernetes 服务账户相关联，用于管理应用程序的证书。要了解有关亚马逊 EKS Pod 身份的更多信息，请参阅亚马逊EKS用户指南中的亚马逊 EKS Pod 身份。配置后，Amazon会EKS设置AWS_CONTAINER_CREDENTIALS_FULL_URI和工具用来获取凭证的SDKs和AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE环境变量。有关设置信息，请参阅亚马逊EKS用户指南中的设置 Amazon EKS Pod 身份代理，或者在 AWS 博客网站上的 Amazon EKS Pod Identity 简化了亚马逊EKS集群上应用程序的IAM权限。

使用以下方法配置此功能：

AWS_CONTAINER_CREDENTIALS_FULL_URI - 环境变量

指定在请求凭证时SDK要使用的完整HTTPURL终端节点。这包括方案和主机。

默认值：无。

有效值：有效URI。

注意：此设置是 *AWS_CONTAINER_CREDENTIALS_RELATIVE_URI* 的替代设置，只有在未设置 *AWS_CONTAINER_CREDENTIALS_RELATIVE_URI* 时才会使用。

Linux/macOS 通过命令行设置环境变量的示例：

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

或者

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

AWS_CONTAINER_CREDENTIALS_RELATIVE_URI - 环境变量

指定在请求凭证时SDK要使用的相对HTTPURL端点。该值将附加到默认的 Amazon ECS 主机名169.254.170.2中。

默认值：无。

有效值：相对有效URI。

Linux/macOS 通过命令行设置环境变量的示例：

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

AWS_CONTAINER_AUTHORIZATION_TOKEN - 环境变量

指定纯文本的授权令牌。如果设置了此变量，则SDK将在HTTP请求上使用环境变量的值设置授权标头。

默认值：无。

有效值：字符串。

注意：此设置是 *AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE* 的替代设置，只有在未设置 *AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE* 时才会使用。

Linux/macOS 通过命令行设置环境变量的示例：

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

AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE - 环境变量

指定至包含纯文本授权令牌的文件的绝对文件路径。

默认值：无。

有效值：字符串。

Linux/macOS 通过命令行设置环境变量的示例：

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

兼容 AWS SDKs

以下内容SDKs支持本主题中描述的功能和设置。所有部分例外情况均已注明。 AWS SDK for Java 和 AWS SDK for Kotlin 唯一支持任何JVM系统属性设置。

SDK	支持	备注或更多信息
AWS CLI v2	是	
SDK对于 C++	是	

SDK	支持	备注或更多信息
SDK适用于 Go V2 (1.x)	是	
SDK适用于 Go 1.x (V1)	是	
SDK适用于 Java 2.x	是	AWS_CONTAINER_CREDENTIALS_FULL_URI AWS_CONTAINER_AUTHORIZATION_TOKEN 也用于适用于 Java 的 Lambda SnapStart 。
SDK适用于 Java 1.x	是	AWS_CONTAINER_CREDENTIALS_FULL_URI AWS_CONTAINER_AUTHORIZATION_TOKEN 也用于适用于 Java 的 Lambda SnapStart 。
SDK适用于 JavaScript 3.x	是	
SDK适用于 JavaScript 2.x	是	
SDK对于 Kotlin 来说	是	
SDK对于。 NET3.x	是	
SDK适用于 PHP 3.x	是	
SDK适用于 Python (Boto3)	是	
SDK适用于 Ruby 3.x	是	
SDK对于 Rust	是	
SDK为斯威夫特	是	
用于 PowerShell	是	

IAM身份中心凭证提供商

此身份验证机制 AWS IAM Identity Center 用于获取您的代码的单点登录 (SSO) 访问 AWS 服务 权限。

> **Note**
>
> 在 AWS SDKAPI文档中，IAM身份中心凭据提供者被称为凭SSO证提供者。

启用 IAM Identity Center 后，您可以在共享 AWS `config`文件中为其设置定义配置文件。此配置文件用于连接到IAM身份中心访问门户。当用户成功通过 Identity Center 进行IAM身份验证后，门户会返回与该用户关联的IAM角色的短期证书。要了解如何从配置中SDK获取临时证书并将其用于 AWS 服务 请求，请参阅了解 IAM Identity Center 身份验证。

通过该`config`文件配置IAM身份中心有两种方法：

- （推荐）SSO令牌提供者配置-延长会话持续时间。包括对自定义会话持续时间的支持。
- 传统不可刷新的配置-使用固定的八小时会话。

在这两种配置中，您都需要在会话到期后重新登录。

以下两份指南包含有关IAM身份中心的其他信息：

- AWS IAM Identity Center 用户指南
- AWS IAM Identity Center 门户网站API参考

要深入了解SDKs和工具如何使用和使用此配置刷新凭据，请参阅了解 IAM Identity Center 身份验证。

先决条件

您必须先启用 “IAM身份中心”。有关启用 Ident IAM ity Center 身份验证的详细信息，请参阅AWS IAM Identity Center 用户指南 AWS IAM Identity Center中的启用。

> **Note**
>
> 或者，有关本页详细介绍的完整先决条件和必要的共享`config`文件配置，请参阅设置指导说明IAM您的SDK或工具的身份中心身份验证。

SSO令牌提供者配置

当您使用SSO令牌提供程序配置时，您的 AWS SDK或工具会自动刷新您的会话，直到延长的会话时段为止。有关会话持续时间和最长持续时间的更多信息，请参阅AWS IAM Identity Center 用户指南中的配置 AWS 访问门户和 Ident IAM ity Center 集成应用程序的会话持续时间。

该config文件的sso-session部分用于对用于获取SSO访问令牌的配置变量进行分组，然后可以使用这些变量来获取 AWS 凭证。有关config文件中此部分的更多详细信息，请参阅配置文件的格式。

以下共享config文件示例使用配置文件配置SDK或工具，以请求 Ident dev IAM ity Center 凭证。

```
[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
```

前面的示例显示了您定义一个sso-session截面并将其与截面相关联。通常，sso_role_name必须在profile部分中设置sso_account_id和，以便SDK可以请求 AWS 证书。sso_regionsso_start_url、和，sso_registration_scopes必须在该sso-session部分中设置。

sso_account_id并且sso_role_name不是所有SSO令牌配置场景都必需的。如果您的应用程序仅使用支持持 AWS 服务 有者身份验证的凭证，则不需要传统 AWS 凭证。持有人身份验证是一种使用称为不记名令牌的安全令牌的HTTP身份验证方案。在这种情况下，不需要 sso_account_id 和 sso_role_name。要确定该服务是否支持不记名令牌授权，请参阅个人 AWS 服务 指南。

注册范围配置为 sso-session 的一部分。作用域是一种机制 OAuth 2.0 限制应用程序对用户帐户的访问权限。前面的示例设置sso_registration_scopes为列出账户和角色提供必要的访问权限。

以下示例显示了如何在多个配置文件中重复使用相同的sso-session配置。

```
[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
```

身份验证令牌缓存到 ~/.aws/sso/cache 目录下的磁盘上，文件名基于会话名称。

遗留的不可刷新配置

使用遗留的不可刷新配置不支持自动令牌刷新。我们建议改用 SSO令牌提供者配置。

要使用传统的不可刷新配置，您必须在配置文件中指定以下设置：

- sso_start_url
- sso_region
- sso_account_id
- sso_role_name

可以使用 sso_start_url 和 sso_region 设置为配置文件指定用户门户。可以使用 sso_account_id 和 sso_role_name 设置来指定权限。

以下示例设置了 config 文件中的四个必需值。

```
[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
```

身份验证令牌缓存到 ~/.aws/sso/cache 目录下的磁盘上，文件名基于 sso_start_url。

IAM身份中心凭证提供商设置

使用以下方法配置此功能：

sso_start_url-共享 AWS **config**文件设置

指向贵组织的 Ident IAM ity Center 颁发者URL或访问门户URL的。URL有关更多信息，请参阅《AWS IAM Identity Center 用户指南》中的使用 AWS 访问门户。

要找到此值，请打开 Ident IAMity Center 控制台，查看控制面板，找到AWS 访问门户URL。

- 或者，从的 2.22.0 版本开始 AWS CLI，您可以改用发行者的值。AWS URL

sso_region-共享 AWS **config**文件设置

其中 AWS 区域 包含您的IAM身份中心门户主机；也就是您在启用 Ident IAM ity Center 之前选择的区域。这与您的默认 AWS 区域无关，也可能有所不同。

有关 AWS 区域 及其代码的完整列表，请参阅中的区域终端节点Amazon Web Services 一般参考。要查找此值，请打开 Ident IAMity Center 控制台，查看控制面板，然后找到区域。

sso_account_id-共享 AWS **config**文件设置

通过 AWS Organizations 服务添加 AWS 账户 的用于身份验证的数字 ID。

要查看可用账户列表，请转到IAM身份中心控制台并打开AWS 账户页面。您还可以使用AWS IAM Identity Center 门户API参考中的ListAccountsAPI方法查看可用账户列表。例如，您可以调用 “列表账户” AWS CLI 方法。

sso_role_name-共享 AWS **config**文件设置

作为IAM角色配置的权限集的名称，用于定义用户生成的权限。角色必须存在于 AWS 账户 指定的中`sso_account_id`。使用角色名称，而不是角色 Amazon 资源名称 (ARN)。

权限集附有IAM策略和自定义权限策略，并定义了用户对其分配的访问权限级别 AWS 账户。

要查看每个可用权限集的列表 AWS 账户，请转到 Ident IAMity Center 控制台并打开AWS 账户页面。选择 AWS 账户 表格中列出的正确权限集名称。您还可以使用AWS IAM Identity Center 门户API参考中的ListAccountRolesAPI方法查看可用权限集列表。例如，您可以调用 AWS CLI 方法list-account-roles。

sso_registration_scopes-共享 AWS **config**文件设置

要为 `sso-session` 授权的范围的逗号分隔列表。应用程序可以请求一个或多个范围，向应用程序签发的访问令牌将仅限于授予的范围。要从 Ident IAM ity Center 服务取回刷新令牌，`sso:account:access`必须授予最小范围。有关可用访问范围选项的列表，请参阅AWS IAM Identity Center 用户指南中的访问范围。

这些范围定义了为注册OIDC客户端请求授权的权限以及客户端检索的访问令牌。作用域授权访问IAM身份中心持有者令牌授权的端点。

此设置不适用于遗留的不可刷新配置。使用传统配置发布的令牌被隐式限制在 `sso:account:access` 作用域范围内。

兼容 AWS SDKs

以下内容SDKs支持本主题中描述的功能和设置。所有部分例外情况均已注明。 AWS SDK for Java 和 AWS SDK for Kotlin 唯一支持任何JVM系统属性设置。

SDK	支持	备注或更多信息
AWS CLI v2	是	
SDK对于 C++	是	
SDK适用于 Go V2 (1.x)	是	
SDK适用于 Go 1.x (V1)	是	要使用共享 `config` 文件设置，必须开启从配置文件加载的功能；请参阅会话。
SDK适用于 Java 2.x	是	`credentials` 文件中也支持配置值。
SDK适用于 Java 1.x	否	
SDK适用于 JavaScript 3.x	是	
SDK适用于 JavaScript 2.x	是	
SDK对于 Kotlin 来说	是	
SDK对于。 NET3.x	是	
SDK适用于 PHP 3.x	是	
SDK适用于 Python (Boto3)	是	
SDK适用于 Ruby 3.x	是	

SDK	支持	备注或更多信息
SDK对于 Rust	部分	仅限遗留的不可刷新配置。
SDK为斯威夫特	是	
用于 PowerShell	是	

IMDS凭证提供商

实例元数据服务 (IMDS) 提供有关您的实例的数据，您可以使用这些数据来配置或管理正在运行的实例。有关可用数据的更多信息，请参阅 Amazon EC2 用户指南中的使用实例元数据。Amazon EC2 提供了可供实例使用的本地终端节点，该终端节点可以为实例提供各种信息。如果实例附加了角色，则它可以提供一组对该角色有效的凭证。SDKs可以使用该端点来解析作为其默认凭证提供程序链一部分的证书。默认使用实例元数据服务版本 2 (IMDSv2)，IMDS这是使用会话令牌的更安全的版本。如果由于不可重试的情况（HTTP错误代码 403、404、405）而失败，则用作后备。IMDSv1

使用以下方法配置此功能：

`AWS_EC2_METADATA_DISABLED` - 环境变量

是否尝试使用 Amazon EC2 实例元数据服务 (IMDS) 来获取证书。

默认值：`false`。

有效值：

- **`true`**— 请勿使用IMDS来获取证书。
- **`false`**— 用于IMDS获取凭证。

`ec2_metadata_v1_disabled`-共享 AWS **`config`**文件设置, **`AWS_EC2_METADATA_V1_DISABLED`** - 环境变量, **`aws.disableEc2MetadataV1`**-JVM 系统属性：仅限 Java/Kotlin

如果IMDSv2失败，是否使用实例元数据服务版本 1 (IMDSv1) 作为后备方案。

> **Note**
>
> New SDKs 不支持IMDSv1，因此不支持此设置。有关详细信息，请见表 兼容 AWS SDKs。

默认值：`false`。

有效值：

- **`true`**— 请勿IMDSv1用作备用。
- **`false`**— IMDSv1 用作备用。

`ec2_metadata_service_endpoint`-共享 AWS **`config`**文件设置, **`AWS_EC2_METADATA_SERVICE_ENDPOINT`** - 环境变量, **`aws.ec2MetadataServiceEndpoint`**-JVM 系统属性：仅限 Java/Kotlin

的终端节点IMDS。此值将覆盖 AWS SDKs和工具搜索亚马逊EC2实例元数据的默认位置。

默认值：如果 `ec2_metadata_service_endpoint_mode` 等于 `IPv4`，则默认端点为 `http://169.254.169.254`。如果 `ec2_metadata_service_endpoint_mode` 等于 `IPv6`，则默认端点为 `http://[fd00:ec2::254]`。

有效值：有效URI。

`ec2_metadata_service_endpoint_mode`-共享 AWS **`config`**文件设置, **`AWS_EC2_METADATA_SERVICE_ENDPOINT_MODE`** - 环境变量, **`aws.ec2MetadataServiceEndpointMode`**-JVM 系统属性：仅限 Java/Kotlin

的终端节点模式IMDS。

默认值：`IPv4`。

有效值：`IPv4`、`IPv6`。

> **Note**
>
> IMDS证书提供者是其中的一部分。了解凭证提供商链但是，只有在本系列中的其他几个提供商之后，才会对IMDS证书提供者进行检查。因此，如果您希望您的程序使用此提供者的凭证，则必须从配置中删除其他有效的凭证提供者或使用其他配置文件。或者，与其依赖凭证提供者链

来自动发现哪个提供者返回了有效凭证，不如在代码中指定证书提供者的用法。IMDS创建服务客户端时，可直接指定凭证来源。

IMDS证书的安全性

默认情况下，如果未配置有效的证书， AWS SDK则SDK会尝试使用亚马逊EC2实例元数据服务 (IMDS) 来检索 AWS 角色的证书。通过将 `AWS_EC2_METADATA_DISABLED` 环境变量设置为 `true`，可以禁用此行为。这样可以防止不必要的网络活动，并增强可能被模仿 Amazon EC2 实例元数据服务的不可信网络的安全性。

> **Note**
>
> AWS SDK无论这些设置如何，配置了有效凭据的客户端都不会使用IMDS来检索凭据。

禁用亚马逊EC2IMDS凭证的使用

如何设置此环境变量取决于所使用的操作系统以及您是否希望更改保持不变。

Linux 和 macOS

使用 Linux 或 macOS 的客户可以使用以下命令设置此环境变量：

```
$ export AWS_EC2_METADATA_DISABLED=true
```

如果您希望此设置在多个 shell 会话和系统重启中保持不变，则可以将上述命令添加到您的 shell 配置文件中，例如 `.bash_profile`、`.zsh_profile` 或 `.profile`。

Windows

使用 Windows 的客户可以使用以下命令设置此环境变量：

```
$ set AWS_EC2_METADATA_DISABLED=true
```

如果您希望此设置在多个 shell 会话和系统重启中保持不变，则可以改用以下命令：

```
$ setx AWS_EC2_METADATA_DISABLED=true
```

> **Note**
>
> 该 setx 命令不会将该值应用于当前的 shell 会话，因此您需要重新加载或重新打开 shell 才能使更改生效。

兼容 AWS SDKs

以下内容SDKs支持本主题中描述的功能和设置。所有部分例外情况均已注明。 AWS SDK for Java 和 AWS SDK for Kotlin 唯一支持任何JVM系统属性设置。

SDK	支持	备注或更多信息
AWS CLI v2	是	
SDK对于 C++	是	
SDK适用于 Go V2 (1.x)	是	
SDK适用于 Go 1.x (V1)	是	要使用共享 config 文件设置，必须开启从配置文件加载的功能；请参阅会话。
SDK适用于 Java 2.x	是	
SDK适用于 Java 1.x	部分	JVM系统属性：使用com.amazonaws.sdk.disableEc2MetadataV1 代替aws.disableEc2MetadataV1 ；aws.ec2MetadataServiceEndpoint 且aws.ec2MetadataServiceEndpointMode 不支持。
SDK适用于 JavaScript 3.x	是	
SDK适用于 JavaScript 2.x	是	
SDK对于 Kotlin 来说	是	不使用IMDSv1后备。
SDK对于。 NET3.x	是	
SDK适用于 PHP 3.x	是	

SDK	支持	备注或更多信息
SDK适用于 Python (Boto3)	是	
SDK适用于 Ruby 3.x	是	
SDK对于 Rust	是	不使用IMDSv1后备。
SDK为斯威夫特	是	
用于 PowerShell	是	您可以使用在代码中显式禁用IMDSv1回退。`[Amazon.Util.EC2InstanceMetadata]::EC2MetadataV1Disabled = $true`

进程凭证提供者

SDKs提供一种针对自定义用例扩展凭证提供者链的方法。此提供程序可用于提供自定义实现，例如从本地凭证存储中检索凭证或与本地身份提供商集成。

例如，Ro IAM les Anywhere 使用`credential_process`来代表您的应用程序获取临时证书。要对此用途配置 `credential_process`，请参阅 IAM Roles Anywhere。

> **Note**
>
> 以下内容描述了一种从外部进程获取凭据的方法，如果您在外部进程之外运行软件，则可以使用该方法。 AWS。 如果你正在建造 AWS 计算资源，请使用其他凭据提供程序。如果使用此选项，则应确保使用操作系统的安全最佳实践尽可能锁定配置文件。确认您的自定义凭证工具未向写入任何机密信息`StdErr`，因为和 SDKs AWS CLI 可以捕获和记录此类信息，从而有可能将其暴露给未经授权的用户。

使用以下方法配置此功能：

`credential_process`-共享 AWS **`config`**文件设置

指定SDK或工具代表您运行的外部命令，以生成或检索要使用的身份验证凭据。该设置指定了SDK将要调用的程序/命令的名称。当SDK调用该进程时，它会等待进程向其写入JSON数

据。stdout自定义提供者必须以特定格式返回信息。该信息包含SDK或工具可用于对您进行身份验证的凭据。

> **Note**
>
> 进程凭证提供者是 了解凭证提供商链 的一部分。但是，只有在本系列中的其他几个提供者之后，才会检查进程凭证提供者。因此，如果您希望您的程序使用此提供者的凭证，则必须从配置中删除其他有效的凭证提供者或使用其他配置文件。或者，与其依赖凭证提供者链自动发现哪个提供者返回了有效凭证，不如在代码中指定使用的进程凭证提供者。创建服务客户端时，可直接指定凭证来源。

指定凭证程序的路径

该设置的值是一个字符串，其中包含SDK或开发工具代表您运行的程序的路径：

- 路径和文件名只能由以下字符组成：A-Z、a-z、0-9、连字符（-）、下划线（_）、句点（.）、正斜杠（/）、反斜杠（\）和空格。
- 如果路径或文件名包含空格，请将完整路径和文件名用双引号 (" ") 括起来。
- 如果参数名称或参数值包含空格，则用双引号 (" ") 将该元素括起来。仅括起来名称或值，而不是名称值对。
- 请勿在字符串中包含任何环境变量。例如，您不能包含 $HOME 或 %USERPROFILE%。
- 不要将主文件夹指定为 ~。* 您必须指定完整路径或基文件名。如果存在基本文件名，则系统会尝试在 PATH 环境变量指定的文件夹中查找该程序。路径因操作系统而异：

  以下示例显示了在 Linux/macOS 上的共享 config 文件中设置 credential_process。

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

  以下示例显示了在 Windows 上的共享 config 文件中设置 credential_process。

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

- 可以在专用配置文件中指定：

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

凭证计划的有效输出

SDK运行配置文件中指定的命令，然后从标准输出流中读取数据。无论是脚本还是二进制程序，您指定的命令都必须生成STDOUT符合以下语法的JSON输出。

```
{
    "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**
>
> 截至撰写本文之时，`Version` 密钥必须设置为 1。随时间推移和该结构的发展，该值可能会增加。

密`Expiration`钥是一个RFC3339格式化的时间戳。如果该`Expiration`密钥未出现在工具的输出中，则SDK假设这些证书是不会刷新的长期凭证。否则，将其视为临时凭证，并通过在其过期前重新运行 `credential_process` 命令来自动刷新凭证。

> **Note**
>
> SDK不会像承担角色凭据那样缓存外部进程凭证。如果需要缓存，则必须在外部进程中实现。

外部进程可以返回非零返回代码，以指示在检索凭证时发生错误。

兼容 AWS SDKs

以下内容SDKs支持本主题中描述的功能和设置。所有部分例外情况均已注明。支持任何JVM系统属性设置 AWS SDK for Java 还有 AWS SDK for Kotlin 只有。

SDK	支持	备注或更多信息
AWS CLI v2	是	
SDK对于 C++	是	
SDK适用于 Go V2 (1.x)	是	
SDK适用于 Go 1.x (V1)	是	要使用共享 `config` 文件设置，必须开启从配置文件加载的功能；请参阅会话。
SDK适用于 Java 2.x	是	
SDK适用于 Java 1.x	是	
SDK适用于 JavaScript 3.x	是	
SDK适用于 JavaScript 2.x	是	
SDK对于 Kotlin 来说	是	
SDK对于。 NET3.x	是	
SDK适用于 PHP 3.x	是	
SDK适用于 Python (Boto3)	是	
SDK适用于 Ruby 3.x	是	
SDK对于 Rust	是	
SDK为斯威夫特	是	
用于 PowerShell	是	

AWS SDKs和 “工具” 标准化功能

许多功能已标准化为一致的默认值，并且在许多功能上都以相同的方式工作SDKs。这种一致性可以提高跨多个编码时的生产力和清晰度SDKs。所有设置都可以在代码中被覆盖，有关详细信息，请参阅您的具体设置SDKAPI。

> **Important**
>
> 并非所有功能都SDKs支持所有功能，甚至不是功能中的所有方面。

主题

基于账户的终端节点

基于账户的终端节点使用您的 AWS 账户 ID 来简化支持此功能的服务 AWS 服务 请求的路由，从而帮助确保高性能和可扩展性。当您使用支持基于账户的 AWS SDK终端节点的凭证提供商和服务

时，SDK将自动构建和使用基于账户的终端节点，而不是区域终端节点。基于账户的终端节点的形式为`https://<account-id>.ddb.<region>.amazonaws.com`，其中替换`<account-id>`为你的 AWS 账户 ID，替换`<region>`为你的 ID AWS 区域

默认情况下，账户 ID 是在处理请求时收集的，并用于构造终端节点。凭证解析也会在处理请求时发生，并且可能会更改端点解析的方法。根据您使用的凭证提供商，账户 ID 的来源可能有所不同。

使用以下方法配置此功能：

aws_account_id-共享 AWS **config**文件设置, **AWS_ACCOUNT_ID** - 环境变量, **aws.accountId**-JVM 系统属性：仅限 Java/Kotlin

AWS 账户 身份证。用于基于账户的端点路由。 AWS 账户 身份证的格式类似于 111122223333。

基于账户的端点路由可为某些服务提供更好的请求性能。

account_id_endpoint_mode-共享 AWS **config**文件设置, **AWS_ACCOUNT_ID_ENDPOINT_MODE** - 环境变量, **aws.accountIdEndpointMode**-JVM 系统属性：仅限 Java/Kotlin

此设置用于在必要时关闭基于账户的端点路由，并绕过基于账户的规则。

默认值：`preferred`

有效值：

- **preferred**— 端点应包括账户 ID（如果有）。
- **disabled**— 已解析的端点不包含账户 ID。
- **required**— 端点必须包含账户 ID。如果账户 ID 不可用，则会SDK引发错误。

兼容 AWS SDKs

以下内容SDKs支持本主题中描述的功能和设置。所有部分例外情况均已注明。 AWS SDK for Java 和 AWS SDK for Kotlin 唯一支持任何JVM系统属性设置。

SDK	支持	SDK版本中发布	备注或更多信息
AWS CLI v2	否		
SDK对于 C++	否		

SDK	支持	SDK版本中发布	备注或更多信息
SDK适用于 Go V2 (1.x)	是	v1.35.0	
SDK适用于 Go 1.x (V1)	否		
SDK适用于 Java 2.x	是	v2.28.4	
SDK适用于 Java 1.x	是	v1.12.771	
SDK适用于 JavaScript 3.x	是	v3.656.0	
SDK适用于 JavaScript 2.x	否		
SDK对于 Kotlin 来说	是	v1.3.37	
SDK对于。 NET3.x	否		
SDK适用于 PHP 3.x	是	v3.318.0	
SDK适用于 Python (Boto3)	否		
SDK适用于 Ruby 3.x	是	v1.123.0	
SDK对于 Rust	否		
SDK为斯威夫特	否		
用于 PowerShell	否		

应用程序 ID

单曲 AWS 账户 可供多个客户应用程序使用来拨打电话 AWS 服务。 应用程序 ID 为客户提供了一种识别哪个源应用程序使用调用的方法 AWS 账户. AWS SDKs而且，服务不会使用或解释此值，除非将其

显示在客户通信中。例如，此值可以包含在操作电子邮件中或在 AWS Health Dashboard 以唯一标识您的哪些应用程序与通知相关联。

使用以下方法配置此功能：

sdk_ua_app_id-共享 AWS **config**文件设置, **AWS_SDK_UA_APP_ID** - 环境变量, **aws.userAgentAppId**-JVM 系统属性：仅限 Java/Kotlin

此设置是您分配给应用程序的唯一字符串，用于识别特定应用程序中的哪些应用程序 AWS 账户 拨打电话 AWS.

默认值：None

有效值：最大长度为 50 的字符串。允许使用字母、数字和以下特殊字符：!、、$、%、&、*、+、-、.、,、^、、_、`、|、~。

在 config 文件中设置此值的示例：

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

Linux/macOS 通过命令行设置环境变量的示例：

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

Windows 通过命令行设置环境变量的示例：

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

如果包含对所用外壳具有特殊含义的符号，请根据需要对该值进行转义。

兼容 AWS SDKs

以下内容SDKs支持本主题中描述的功能和设置。所有部分例外情况均已注明。支持任何JVM系统属性设置 AWS SDK for Java 还有 AWS SDK for Kotlin 只有。

SDK	支持	备注或更多信息
AWS CLI v2	是	
SDK对于 C++	是	不支持共享的config文件。
SDK适用于 Go V2 (1.x)	是	
SDK适用于 Go 1.x (V1)	否	
SDK适用于 Java 2.x	部分	不支持共享config文件设置；不支持环境变量。
SDK适用于 Java 1.x	否	
SDK适用于 JavaScript 3.x	是	
SDK适用于 JavaScript 2.x	否	
SDK对于 Kotlin 来说	是	
SDK对于。 NET3.x	是	不支持环境变量。
SDK适用于 PHP 3.x	是	
SDK适用于 Python (Boto3)	是	
SDK适用于 Ruby 3.x	是	
SDK对于 Rust	是	
SDK为斯威夫特	否	
用于 PowerShell	否	

Amazon EC2 实例元数据

Amazon 在实例上EC2提供一项名为实例元数据服务 (IMDS) 的服务。要了解有关此服务的更多信息，请参阅 Amazon EC2 用户指南中的使用实例元数据。尝试在已配置IAM角色的 Amazon EC2 实例上检索凭证时，与实例元数据服务的连接是可以调整的。

使用以下方法配置此功能：

metadata_service_num_attempts-共享 AWS **config**文件设置, **AWS_METADATA_SERVICE_NUM_ATTEMPTS** - 环境变量

本设置指定了尝试从实例元数据服务检索数据时，在放弃前尝试的总次数。

默认值： 1

有效值：大于或等于 1 的数字。

metadata_service_timeout-共享 AWS **config**文件设置, **AWS_METADATA_SERVICE_TIMEOUT** - 环境变量

指定的从实例元数据服务检索数据时，发生超时前的秒数。

默认值： 1

有效值：大于或等于 1 的数字。

在 config 文件中设置这些值的示例：

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

Linux/macOS 通过命令行设置环境变量的示例：

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

Windows 通过命令行设置环境变量的示例：

```
setx AWS_METADATA_SERVICE_NUM_ATTEMPTS 10
```

```
setx AWS_METADATA_SERVICE_TIMEOUT 10
```

兼容 AWS SDKs

以下内容SDKs支持本主题中描述的功能和设置。所有部分例外情况均已注明。支持任何JVM系统属性设置 AWS SDK for Java 还有 AWS SDK for Kotlin 只有。

SDK	支持	备注或更多信息
AWS CLI v2	是	
SDK对于 C++	否	
SDK适用于 Go V2 (1.x)	否	
SDK适用于 Go 1.x (V1)	否	
SDK适用于 Java 2.x	否	
SDK适用于 Java 1.x	部分	仅支持 AWS_METADATA_SERVICE_TIMEOUT 。
SDK适用于 JavaScript 3.x	否	
SDK适用于 JavaScript 2.x	否	
SDK对于 Kotlin 来说	否	
SDK对于。 NET3.x	否	
SDK适用于 PHP 3.x	是	
SDK适用于 Python (Boto3)	是	
SDK适用于 Ruby 3.x	否	
SDK对于 Rust	否	
SDK为斯威夫特	否	

SDK	支持	备注或更多信息
用于 PowerShell	否	

Amazon S3 接入点

Amazon S3 服务提供接入点作为与 Amazon S3 存储桶交互的替代方式。接入点上可以应用唯一的策略和配置，而不是直接应用到存储桶。与 AWS SDKs，您可以在存储桶字段中使用访问点 Amazon 资源名称 (ARNs) 进行API操作，而不必明确指定存储桶名称。它们用于特定的操作，例如使用访问点ARN从存储桶中获取对象，或使用访问点ARN将对象`PutObject`添加到存储桶。`GetObject`

要了解有关 Amazon S3 接入点的更多信息ARNs，请参阅 Amazon S3 用户指南中的使用接入点。

使用以下方法配置此功能：

`s3_use_arn_region`-共享 AWS **`config`**文件设置, **`AWS_S3_USE_ARN_REGION`** - 环境变量, **`aws.s3UseArnRegion`**-JVM 系统属性：仅限 Java/Kotlin, 要直接在代码中配置值，请SDK直接咨询您的具体内容。

> 此设置控制是否SDK使用接入点 ARN AWS 区域 为请求构建区域终端节点。SDK验证了 ARN AWS 区域 由相同的方式提供 AWS 按照客户端的配置进行分区 AWS 区域 以防止最有可能失败的跨分区调用。如果多次定义，则优先使用代码配置的设置，其次是环境变量设置。
>
> 默认值：`false`
>
> 有效值：
>
> - **`true`**— SDK 用途 t ARN he's AWS 区域 构造端点而不是客户端配置的端点时 AWS 区域。 例外：如果已配置客户端 AWS 区域 是一个 FIPS AWS 区域，那么它必须与 ARN's 匹配 AWS 区域。 否则，将导致错误。
> - **`false`**— SDK 使用客户端的配置 AWS 区域 在构造端点时。

兼容 AWS SDKs

以下内容SDKs支持本主题中描述的功能和设置。所有部分例外情况均已注明。支持任何JVM系统属性设置 AWS SDK for Java 还有 AWS SDK for Kotlin 只有。

SDK	支持	备注或更多信息
AWS CLI v2	是	
SDK对于 C++	是	
SDK适用于 Go V2 (1.x)	是	
SDK适用于 Go 1.x (V1)	是	要使用共享 config 文件设置，必须开启从配置文件加载的功能；请参阅会话。
SDK适用于 Java 2.x	是	
SDK适用于 Java 1.x	是	JVM不支持系统属性。
SDK适用于 JavaScript 3.x	是	
SDK适用于 JavaScript 2.x	是	
SDK对于 Kotlin 来说	是	
SDK对于。 NET3.x	是	不遵循标准优先级；共享的config文件值优先于环境变量。
SDK适用于 PHP 3.x	是	
SDK适用于 Python (Boto3)	是	
SDK适用于 Ruby 3.x	是	
SDK对于 Rust	否	
SDK为斯威夫特	否	
用于 PowerShell	是	不遵循标准优先级；共享的config文件值优先于环境变量。

Amazon S3 多区域访问点

Amazon S3 多区域接入点提供了一个全局终端节点，应用程序可以使用该终端节点来满足来自位于多个 Amazon S3 存储桶的请求 AWS 区域。 您可以使用多区域接入点来构建具有与单个区域相同的架构的多区域应用程序，然后在世界任何地方运行这些应用程序。

要了解有关多区域接入点的更多信息，请参阅 Amazon S3 用户指南中的 Amazon S3 中的多区域接入点。

要了解有关多区域接入点 Amazon 资源名称 (ARNs) 的更多信息，请参阅 Amaz on S3 用户指南中的使用多区域接入点发出请求。

要了解有关创建多区域接入点的更多信息，请参阅 Amazon S3 用户指南中的管理多区域接入点。

SigV4A 算法是用于签署全局区域请求的签名实现。该算法是SDK通过依赖来获得的AWS 常用运行时 (CRT) 库。

使用以下方法配置此功能：

`s3_disable_multiregion_access_points`-共享 AWS **`config`**文件设置, **`AWS_S3_DISABLE_MULTIREGION_ACCESS_POINTS`** - 环境变量, **`aws.s3DisableMultiRegionAccessPoints`**-JVM 系统属性：仅限 Java/Kotlin, 要直接在代码中配置值，请SDK直接咨询您的具体内容。

此设置可控制是否SDK可能尝试跨区域请求。如果多次定义，则优先使用代码配置的设置，其次是环境变量设置。

默认值：`false`

有效值：

- **`true`** – 停止使用跨区域请求。
- **`false`** – 使用多区域接入点启用跨区域请求。

兼容 AWS SDKs

以下内容SDKs支持本主题中描述的功能和设置。所有部分例外情况均已注明。支持任何JVM系统属性设置 AWS SDK for Java 还有 AWS SDK for Kotlin 只有。

SDK	支持	备注或更多信息
AWS CLI v2	是	
SDK对于 C++	是	
SDK适用于 Go V2 (1.x)	是	
SDK适用于 Go 1.x (V1)	否	
SDK适用于 Java 2.x	是	
SDK适用于 Java 1.x	否	
SDK适用于 JavaScript 3.x	是	
SDK适用于 JavaScript 2.x	否	
SDK对于 Kotlin 来说	是	
SDK对于。 NET3.x	是	
SDK适用于 PHP 3.x	是	
SDK适用于 Python (Boto3)	是	
SDK适用于 Ruby 3.x	是	
SDK对于 Rust	是	
SDK为斯威夫特	否	
用于 PowerShell	是	

AWS 区域

AWS 区域 是使用时需要理解的重要概念 AWS 服务。

使用 AWS 区域，您可以访问 AWS 服务 实际居住在特定地理区域的内容。这可用于保证您的数据和应用程序接近您和用户访问它们的位置。区域提供容错能力、稳定性和弹性，还可以减少延迟。使用区域，您能够创建保持可用且不受区域中断影响的冗余资源。

大多数 AWS 服务 请求都与特定的地理区域相关联。除非您明确使用 AWS 服务提供的复制功能，否则在一个区域中创建的资源在任何其他区域中都不存在。例如，Amazon S3 和亚马逊EC2支持跨区域复制。某些服务（IAM例如）没有区域资源。

AWS 一般参考 包含有关以下内容的信息：

- 要了解区域和端点之间的关系，并查看现有区域端点的列表，请参阅AWS 服务端点。
- 要查看当前各 AWS 服务所有支持的区域和端点列表，请参阅服务端点和限额。

创建服务客户端

要以编程方式访问 AWS 服务，SDKs请为每个使用客户端类/对象。 AWS 服务例如EC2，如果您的应用程序需要访问亚马逊，则您的应用程序将创建一个 Amazon EC2 客户端对象来与该服务接口。

如果在代码本身中没有为客户端明确指定区域，则客户端将默认使用通过以下设置`region`设置的区域。但是，可以为任何单个客户端对象显式设置客户端的活动区域。以这种方式设置区域优先于该特定服务客户端的任何全局设置。备用区域是在该客户端的实例化过程中指定的，该区域特定于您的客户端SDK（请查看您的特定SDK指南或您的SDK代码库）。

使用以下方法配置此功能：

`region`-共享 AWS **`config`**文件设置, **`AWS_REGION`** - 环境变量, **`aws.region`**-JVM 系统属性：仅限 Java/Kotlin

指定 AWS 请求 AWS 区域 使用的默认值。此区域用于未提供特定区域的SDK服务请求。

默认值：无。必须明确指定此值。

有效值：

- 可用于所选服务的任何区域代码，有关列表，请参阅 AWS 一般参考中的 AWS 服务端点。例如，值 `us-east-1` 将端点设置为 AWS 区域 美国东部（弗吉尼亚州北部）。
- `aws-global`为除区域终端节点之外还支持单独的全局终端节点的服务指定全局终端节点，例如 AWS Security Token Service (AWS STS) 和亚马逊简单存储服务 (Amazon S3) Service。

在 `config` 文件中设置此值的示例：

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

Linux/macOS 通过命令行设置环境变量的示例：

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

Windows 通过命令行设置环境变量的示例：

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

大多数SDKs都有一个 “配置” 对象，可用于在应用程序代码中设置默认区域。有关详细信息，请参阅您的特定 AWS SDK开发者指南。

与之兼容 AWS SDKs

以下内容SDKs支持本主题中描述的功能和设置。所有部分例外情况均已注明。 AWS SDK for Java 和 AWS SDK for Kotlin 唯一支持任何JVM系统属性设置。

SDK	支持	备注或更多信息
AWS CLI v2	是	AWS CLI v2 在中的任何值`AWS_REGION` 之前使用中的任何值 `AWS_DEFAULT_REGION` （两个变量都被选中）。
AWS CLI v1	是	AWS CLI v1 使用`AWS_DEFAULT_REGION` 为此目的命名的环境变量。
SDK对于 C++	是	
SDK适用于 Go V2 (1.x)	是	
SDK适用于 Go 1.x (V1)	是	要使用共享 `config` 文件设置，必须开启从配置文件加载的功能；请参阅会话。
SDK适用于 Java 2.x	是	
SDK适用于 Java 1.x	是	

SDK	支持	备注或更多信息
SDK适用于 JavaScript 3.x	是	
SDK适用于 JavaScript 2.x	是	
SDK对于 Kotlin 来说	是	
SDK对于。 NET3.x	是	
SDK适用于 PHP 3.x	是	
SDK适用于 Python (Boto3)	是	SDK它使用为此目的命AWS_DEFAULT_REGION 名的环境变量。
SDK适用于 Ruby 3.x	是	
SDK对于 Rust	是	
SDK为斯威夫特	是	
用于 PowerShell	是	

AWS STS 区域终端节点

AWS Security Token Service (AWS STS) 既可作为全球服务，也可作为区域服务提供。其中一些AWS SDKs和默认CLIs使用全球服务终端节点 (`https://sts.amazonaws.com`)，而有些则使用区域服务终端节点 (`https://sts.{region_identifier}.{partition_domain}`)。全球请求映射到美国东部（弗吉尼亚北部）区域。有关 AWS STS 终端节点的更多信息，请参阅AWS Security Token Service API参考中的终端节点。或者，AWS STS 在《AWS Identity and Access Management 用户指南》 AWS 区域中学习管理。

AWS 最佳做法是尽可能使用区域终端节点并配置您的终端节点AWS 区域。非商业分区的客户必须使用区域终端节点。并非所有SDKs工具都支持此设置，但所有工具都围绕全球和区域端点定义了行为。有关更多信息，请参阅下文的小节。

对于支持此设置的SDKs和工具，客户可以使用以下方式配置功能：

sts_regional_endpoints-共享 AWS **config**文件设置, **AWS_STS_REGIONAL_ENDPOINTS** - 环境变量

此设置指定SDK或工具如何确定用于与 AWS Security Token Service (AWS STS) 通信的 AWS 服务 端点。

默认值：`legacy`

> **Note**
>
> 2022 年 7 月之后发布的所有新的SDK主要版本都将默认为`regional`。新的SDK主要版本可能会删除此设置并使用`regional`行为。为了减少此变更对未来的影响，我们建议您尽可能在应用程序中开始使用`regional`。

有效值：（建议的值：*`regional`*）

- **`legacy`**— 使用全局 AWS STS 终端节点`sts.amazonaws.com`。
- **`regional`**— SDK 或工具始终使用当前配置区域的 AWS STS 终端节点。例如，如果将客户端配置为使用`us-west-2`，则对的所有调用都将 AWS STS 发送到区域终端节点`sts.us-west-2.amazonaws.com`，而不是全球`sts.amazonaws.com`终端节点。要在启用此设置时向全局终端节点发送请求，您可以将区域设置为 `aws-global`。

在 `config` 文件中设置这些值的示例：

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

Linux/macOS 通过命令行设置环境变量的示例：

```
export AWS_STS_REGIONAL_ENDPOINTS=regional
```

Windows 通过命令行设置环境变量的示例：

```
setx AWS_STS_REGIONAL_ENDPOINTS regional
```

与之兼容 AWS SDKs

> **Note**
>
> AWS 最佳做法是尽可能使用区域终端节点并配置您的终端节点AWS 区域。

下表汇总了您的SDK或工具：

- 支持设置：是否支持STS区域端点的共享config文件变量和环境变量。
- 默认设置值：该设置的默认值（如果支持）。
- 默认服务客户端目标STS端点：即使更改默认端点的设置不可用，客户端也会使用哪个默认端点。
- 服务客户端回退行为：当SDK它应该使用区域终端节点但尚未配置区域时，会怎么做。无论它使用区域终端节点是因为默认还是因为设置选择regional了区域端点，都会出现这种行为。

该表还使用了以下值：

- 全局终端节点:https://sts.amazonaws.com.
- 区域终端节点：基于您的应用程序AWS 区域使用的配置。
- **us-east-1**（区域）：使用**us-east-1**区域终端节点，但会话令牌比典型的全局请求长。

SDK	默认设置值	默认服务客户端目标STS端点	服务客户端回退行为	备注或更多信息
AWS CLI v2	不适用	区域端点	全球终端节点	
AWS CLI v1	legacy	全球终端节点	全球终端节点	
SDK对于 C++	不适用	区域端点	us-east-1（区域）	
SDK适用于 Go V2 (1.x)	不适用	区域端点	请求失败	

SDK		默认设置值	默认服务客户端目标STS端点	服务客户端回退行为	备注或更多信息
SDK适用于 Go 1.x (V1)	是	legacy	全球终端节点	全球终端节点	要使用共享 config 文件设置，必须开启从配置文件加载的功能；请参阅会话。
SDK适用于 Java 2.x	否	不适用	区域端点	请求失败	如果未配置任何区域，则AssumeRole 和AssumeRoleWithWebIdentity 将使用全局STS终端节点
SDK适用于 Java 1.x	是	legacy	全球终端节点	全球终端节点	
SDK适用于 JavaScript 3.x	否	不适用	区域端点	请求失败	
SDK适用于 JavaScript 2.x	是	legacy	全球终端节点	全球终端节点	
SDK对于 Kotlin 来说	否	不适用	区域端点	全球终端节点	
SDK对于。NET3.x	是	legacy	全球终端节点	全球终端节点	
SDK适用于 PHP 3.x	是	legacy	全球终端节点	请求失败	
SDK适用于 Python (Boto3)	是	legacy	全球终端节点	全球终端节点	

SDK		默认设置值	默认服务客户端目标STS端点	服务客户端回退行为	备注或更多信息
SDK适用于Ruby 3.x	是	`regional`	区域端点	请求失败	
SDK对于Rust	否	不适用	区域端点	请求失败	
SDK为斯威夫特	否	不适用	区域端点	请求失败	
用于PowerShell	是	`legacy`	全球终端节点	全球终端节点	

双栈和端点 FIPS

使用以下方法配置此功能：

`use_dualstack_endpoint`-共享 AWS **`config`**文件设置，**`AWS_USE_DUALSTACK_ENDPOINT`** - 环境变量，**`aws.useDualstackEndpoint`**-JVM 系统属性：仅限 Java/Kotlin

开启或关闭SDK是否向双堆栈终端节点发送请求。要了解有关双堆栈终端节点的更多信息，请参阅《亚马逊简单存储服务用户指南》中的 “使用 Amazon S3 双栈终端节点”。IPv4 IPv6双堆栈端点适用于某些区域。

默认值：`false`

有效值：

- **`true`**— SDK 或工具将尝试使用双堆栈端点发出网络请求。如果服务不存在双栈端点和/或 AWS 区域，则请求将失败。
- **`false`**— SDK 或工具不会使用双堆栈端点发出网络请求。

use_fips_endpoint-共享 AWS **config**文件设置, **AWS_USE_FIPS_ENDPOINT** - 环境变量, **aws.useFipsEndpoint**-JVM 系统属性：仅限 Java/Kotlin

开启或关闭SDK或工具是否向FIPS符合标准的端点发送请求。联邦信息处理标准 (FIPS) 是美国政府对数据及其加密的一系列安全要求。政府机构、合作伙伴以及希望与联邦政府做生意的机构必须遵守FIPS指导方针。与标准版不同 AWS 端点，FIPS端点使用符合 FIPS 140-2 的TLS软件库。如果启用了此设置并且您的服务中不存在FIPS终端节点 AWS 区域， AWS 呼叫可能会失败。 特定于服务的端点还有的`--endpoint-url`选项 AWS Command Line Interface 覆盖此设置。

要详细了解通过以下方式指定FIPS终端节点的其他方法 AWS 区域，请参阅按服务划分的FIPS终端节点。有关亚马逊弹性计算云服务终端节点的更多信息，请参阅《亚马逊EC2API参考》中的双栈（IPv4和IPv6）终端节点。

默认值：`false`

有效值：

- **true**— SDK 或工具将向FIPS兼容的端点发送请求。
- **false**— SDK 或工具不会向FIPS符合标准的端点发送请求。

兼容 AWS SDKs

以下内容SDKs支持本主题中描述的功能和设置。所有部分例外情况均已注明。支持任何JVM系统属性设置 AWS SDK for Java 还有 AWS SDK for Kotlin 只有。

SDK	支持	备注或更多信息
AWS CLI v2	是	
SDK对于 C++	是	
SDK适用于 Go V2 (1.x)	是	
SDK适用于 Go 1.x (V1)	是	要使用共享 `config` 文件设置，必须开启从配置文件加载的功能；请参阅会话。
SDK适用于 Java 2.x	是	
SDK适用于 Java 1.x	否	

SDK	支持	备注或更多信息
SDK适用于 JavaScript 3.x	是	
SDK适用于 JavaScript 2.x	是	
SDK对于 Kotlin 来说	是	
SDK对于。 NET3.x	是	
SDK适用于 PHP 3.x	是	
SDK适用于 Python (Boto3)	是	
SDK适用于 Ruby 3.x	是	
SDK对于 Rust	是	
SDK为斯威夫特	是	
用于 PowerShell	是	

端点发现

SDKs使用端点发现来访问服务端点（URLs访问各种资源），同时仍然保持灵活性 AWS 根据需要URLs进行更改。这样，您的代码就可以自动检测新的端点。某些服务没有固定的端点。相反，您可以在运行时通过请求先获取端点来获得可用的端点。检索到可用端点后，代码会使用该端点访问其他操作。例如，对于 Amazon Timestream，SDK会`DescribeEndpoints`请求检索可用的终端节点，然后使用这些终端节点完成特定操作，例如`CreateDatabase`或。`CreateTable`

使用以下方法配置此功能：

`endpoint_discovery_enabled`-共享 AWS **`config`**文件设置，**`AWS_ENABLE_ENDPOINT_DISCOVERY`** - 环境变量，**`aws.endpointDiscoveryEnabled`**-JVM 系统属性：仅限 Java/Kotlin，要直接在代码中配置值，请SDK直接咨询您的具体内容。

开启或关闭 DynamoDB 的终端节点发现。

在 Timestream 中需要发现终端节点，在 Amazon DynamoDB 中是可选的。此设置默认为`true`或，`false`具体取决于服务是否需要端点发现。时间流请求默认为`true`，Amazon DynamoDB 请求默认为。`false`

有效值：

- **`true`**— 对于端点发现是可选的服务，SDK应自动尝试发现终端节点。
- **`false`**— 对于端点发现是可选的服务，SDK不应自动尝试发现终端节点。

与之兼容 AWS SDKs

以下内容SDKs支持本主题中描述的功能和设置。所有部分例外情况均已注明。支持任何JVM系统属性设置 AWS SDK for Java 还有 AWS SDK for Kotlin 只有。

SDK	支持	备注或更多信息
AWS CLI v2	是	
SDK对于 C++	是	
SDK适用于 Go V2 (1.x)	是	
SDK适用于 Go 1.x (V1)	是	要使用共享 `config` 文件设置，必须开启从配置文件加载的功能；请参阅会话。
SDK适用于 Java 2.x	是	SDK适用于 Java 2.x `AWS_ENDPOINT_DISCOVERY_ENABLED` 的，用于环境变量名称。
SDK适用于 Java 1.x	部分	JVM不支持系统属性。
SDK适用于 JavaScript 3.x	是	
SDK适用于 JavaScript 2.x	是	
SDK对于 Kotlin 来说	是	
SDK对于。 NET3.x	是	

SDK	支持	备注或更多信息
SDK适用于 PHP 3.x	是	
SDK适用于 Python (Boto3)	是	
SDK适用于 Ruby 3.x	是	
SDK对于 Rust	部分	仅支持 Timestream。
SDK为斯威夫特	否	
用于 PowerShell	是	

常规配置设置

SDKs支持一些用于配置整体SDK行为的常规设置。

使用以下方法配置此功能：

`api_versions`-共享 AWS **`config`**文件设置

一段时间 AWS 服务维护多个API版本以支持向后兼容。默认情况下，SDK以及 AWS CLI 操作使用最新的可用API版本。要要求使用特定API版本来处理您的请求，请在您的个人资料中添加该`api_versions`设置。

默认值：无。（使用的是最新API版本SDK。）

有效值：这是一个嵌套设置，后面有一行或多行缩进，每行标识一行 AWS 服务和要使用的API版本。有关信息，请参阅文档 AWS 服务以了解有哪些API版本可用。

该示例为两个设置了一个特定的API版本 AWS `config`文件中的服务。这些API版本仅用于在包含这些设置的配置文件下运行的命令。任何其他服务的命令都使用该服务的最新版本API。

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

ca_bundle-共享 AWS **config**文件设置, **AWS_CA_BUNDLE** - 环境变量

指定建立SSL/TLS连接时要使用的自定义证书包（带有.pem扩展名的文件）的路径。

默认值： 无

有效值：指定完整路径或基本文件名。如果存在基本文件名，则系统会尝试在 PATH 环境变量指定的文件夹中查找该程序。

在 config 文件中设置此值的示例：

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

由于操作系统处理路径和路径字符转义的方式存在差异，以下是在 Windows config 文件中设置此值的示例：

```
[default]
ca_bundle = C:\\Users\\username\\.aws\\aws-custom-bundle.pem
```

Linux/macOS 通过命令行设置环境变量的示例：

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

Windows 通过命令行设置环境变量的示例：

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

output-共享 AWS **config**文件设置

指定结果的格式化方式 AWS CLI 和其他 AWS SDKs和工具。

默认值：json

有效值：

- **json**— 输出格式化为字符JSON串。
- **yaml**— 输出格式化为字符YAML串。
- **yaml-stream**— 输出被流式传输并格式化为YAML字符串。串流支持更快地处理大型数据类型。

- **`text`** – 输出采用多个制表符分隔字符串值行的格式。这对于将输出传递到文本处理器（如 `grep`、`sed` 或 `awk`）很有用。
- **`table`** – 输出采用表格形式，使用字符 +|- 以形成单元格边框。它通常以“人性化”格式呈现信息，这种格式比其他格式更容易阅读，但从编程方面来讲不是那么有用。

`parameter_validation`-共享 AWS **`config`**文件设置

指定SDK或工具是否在将命令行参数发送到命令行参数之前尝试对其进行验证 AWS 服务端点。

默认值：`true`

有效值：

- **`true`** – 默认值。SDK或工具对命令行参数执行客户端验证。这有助于SDK或工具确认参数是否有效，并能捕获一些错误。SDK或工具可以在向发送请求之前拒绝无效的请求 AWS 服务端点。
- **`false`**— SDK 或工具在将命令行参数发送到之前不会对其进行验证 AWS 服务端点。这些区域有： AWS 服务端点负责验证所有请求并拒绝无效的请求。

兼容 AWS SDKs

以下内容SDKs支持本主题中描述的功能和设置。所有部分例外情况均已注明。支持任何JVM系统属性设置 AWS SDK for Java 还有 AWS SDK for Kotlin 只有。

SDK	支持	备注或更多信息
AWS CLI v2	部分	`api_versions` 不支持。
SDK对于 C++	是	
SDK适用于 Go V2 (1.x)	部分	不支持 `api_versions` 和 `parameter_validation` 。
SDK适用于 Go 1.x (V1)	部分	不支持 `api_versions` 和 `parameter_validation` 。要使用共享 `config` 文件设置，必须开启从配置文件加载的功能；请参阅会话。
SDK适用于 Java 2.x	否	

SDK	支持	备注或更多信息
SDK适用于 Java 1.x	否	
SDK适用于 JavaScript 3.x	是	
SDK适用于 JavaScript 2.x	是	
SDK对于 Kotlin 来说	否	
SDK对于。 NET3.x	否	
SDK适用于 PHP 3.x	是	
SDK适用于 Python (Boto3)	是	
SDK适用于 Ruby 3.x	是	
SDK对于 Rust	否	
SDK为斯威夫特	否	
用于 PowerShell	否	

IMDS客户端

SDKs使用面向会话的请求实现实例元数据服务版本 2 (IMDSv2) 客户端。有关更多信息IMDSv2，请参阅《Amazon EC2 用户指南》IMDSv2中的 “使用”。IMDS客户端可通过SDK代码库中提供的客户端配置对象进行配置。

使用以下方法配置此功能：

`retries` - 客户端配置对象成员

任何失败的请求的额外重试次数。

默认值：3

有效值：大于 0 的数字。

port - 客户端配置对象成员

端点的端口。

默认值：80

有效值：数字。

token_ttl - 客户端配置对象成员

代币TTL中的那个。

默认值：21,600 秒（6 小时，分配的最长时间)。

有效值：数字。

endpoint - 客户端配置对象成员

的终端节点IMDS。

默认值：如果 endpoint_mode 等于 IPv4，则默认端点为 http://169.254.169.254。如果 endpoint_mode 等于 IPv6，则默认端点为 http://[fd00:ec2::254]。

有效值：有效URI。

大多数人支持以下选项SDKs。有关详细信息，请参阅您的特定SDK代码库。

endpoint_mode - 客户端配置对象成员

的终端节点模式IMDS。

默认值：IPv4

有效值：IPv4、IPv6

http_open_timeout - 客户端配置对象成员（名称可能有所不同）

等待连接打开的秒数。

默认值：1 秒。

有效值：大于 0 的数字。

http_read_timeout - 客户端配置对象成员（名称可能有所不同）

读取一个数据块的秒数。

默认值：1 秒。

有效值：大于 0 的数字。

`http_debug_output` - 客户端配置对象成员（名称可能有所不同）

设置用于调试的输出流。

默认值：无。

有效值：有效的 I/O 流，例如STDOUT。

`backoff` - 客户端配置对象成员（名称可能有所不同）

在两次重试之间休眠的秒数，或者客户提供的回退功能可供调用。这会覆盖默认的指数回退策略。

默认值：因而异SDK。

有效值：因而异SDK。可以是数值，也可以是对自定义函数的调用。

与之兼容 AWS SDKs

以下内容SDKs支持本主题中描述的功能和设置。所有部分例外情况均已注明。 AWS SDK for Java 和 AWS SDK for Kotlin 唯一支持任何JVM系统属性设置。

SDK	支持	备注或更多信息
AWS CLI v2	是	
SDK对于 C++	否	
SDK适用于 Go V2 (1.x)	是	
SDK适用于 Go 1.x (V1)	是	
SDK适用于 Java 2.x	是	
SDK适用于 Java 1.x	是	
SDK适用于 JavaScript 3.x	是	
SDK适用于 JavaScript 2.x	是	

SDK	支持	备注或更多信息
SDK对于 Kotlin 来说	否	
SDK对于。 NET3.x	是	
SDK适用于 PHP 3.x	是	
SDK适用于 Python (Boto3)	是	
SDK适用于 Ruby 3.x	是	
SDK对于 Rust	是	
SDK为斯威夫特	是	
用于 PowerShell	是	

重试行为

重试行为包括有关如何尝试从因向其SDKs发出请求而导致的失败中恢复的设置 AWS 服务.

使用以下方法配置此功能：

retry_mode-共享 AWS **config**文件设置, **AWS_RETRY_MODE** - 环境变量, **aws.retryMode**-JVM 系统属性：仅限 Java/Kotlin

指定SDK或开发者工具如何尝试重试。

默认值：此值特定于您的SDK。请查看您的特定SDK指南或您的SDK代码库以了解其默认值`retry_mode`。

有效值：

- `standard`— （推荐）推荐的重试规则集 AWS SDKs。此模式包括一组标准的重试错误，并自动调整重试次数以最大限度地提高可用性和稳定性。此模式可在多租户应用程序中安全使用。除非`max_attempts`明确配置，否则此模式下默认的最大尝试次数为三次。
- `adaptive`— 一种重试模式，仅适用于特殊用例，包括标准模式的功能以及自动客户端速率限制。除非您注意隔离应用程序租户，否则不建议将此重试模式用于多租户应用程序。请参

阅[在standard和adaptive重试模式之间进行选择](#)了解更多信息。此模式是实验性的，将来可能会改变行为。

- legacy—（不推荐）特定于您的SDK（请查看您的特定SDK指南或您的SDK代码库）。

max_attempts-共享 AWS **config**文件设置, **AWS_MAX_ATTEMPTS** - 环境变量, **aws.maxAttempts**-JVM 系统属性：仅限 Java/Kotlin

指定对请求进行的最大尝试次数。

默认值：如果未指定此值，则其默认值取决于retry_mode设置的值：

- 如果retry_mode是 legacy — 使用您的特定默认值SDK（请查看您的特定SDK指南或代码库以了解max_attempts默认值）。SDK
- 如果retry_mode是standard – 尝试三次。
- 如果retry_mode是adaptive – 尝试三次。

有效值：大于 0 的数字。

在**standard**和**adaptive**重试模式之间进行选择

除非您确定自己的用法更适合，否则我们建议您使用standard重试模式。adaptive

> **Note**
>
> 该adaptive模式假设您正在根据后端服务可能限制请求的范围来池化客户端。如果你不这样做，那么如果你对两个资源使用同一个客户端，那么一个资源中的限制可能会延迟对不相关资源的请求。

Standard	自适应
应用程序用例：全部。	应用程序用例： 1. 对延迟不敏感。 2. 客户机只能访问单个资源，或者，您正在提供逻辑，以便按正在访问的服务资源单独池化客户端。
支持断路以防止在SDK中断期间重试。	支持断路以防止在SDK中断期间重试。

Standard	自适应
在出现故障时使用抖动指数退避。	使用动态退避持续时间来尝试最大限度地减少失败请求的数量，以换取延迟增加的可能性。
永远不要延迟第一次请求尝试，只会延迟重试。	可以限制或延迟初始请求尝试。

如果您选择使用adaptive模式，则您的应用程序必须围绕可能受到限制的每种资源构建客户端。在这种情况下，对资源的调整要比仅仅考虑每种资源都要精细 AWS 服务. AWS 服务 可以有其他维度来限制请求。让我们以亚马逊 DynamoDB 服务为例。DynamoDB 使用 AWS 区域 再加上用于限制请求的访问表。这意味着您的代码正在访问的一个表可能比其他表更受限制。如果您的代码使用同一个客户端访问所有表，并且对其中一个表的请求受到限制，则自适应重试模式将降低所有表的请求速率。您的代码应设计为每个 R egion-and-table 对都有一个客户端。如果您在使用adaptive模式时遇到意外延迟，请参阅具体的 AWS 您正在使用的服务的文档指南。

重试模式实现细节

这些区域有： AWS SDKs使用令牌桶来决定是否应重试请求以及（在adaptive重试模式下）应以多快的速度发送请求。使用两个令牌存储桶SDK：一个重试令牌存储桶和一个请求费率令牌存储桶。

- 重试令牌存储桶用于确定是否SDK应暂时禁用重试，以便在中断期间保护上游和下游服务。在尝试重试之前会从存储桶中获取令牌，请求成功后将令牌返回到存储桶。如果尝试重试时存储桶为空，则SDK不会重试该请求。
- 请求速率令牌存储桶仅在adaptive重试模式下用于确定发送请求的速率。令牌是在发送请求之前从存储桶中获取的，并根据服务返回的限制响应，以动态确定的速率将令牌返回到存储桶。

以下是standard和adaptive两种重试模式的高级伪代码：

```
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)
  }
}
```

以下是关于伪代码中所用组件的更多详细信息：

`GetSendToken`:

此步骤仅在`adaptive`重试模式下使用。此步骤从请求费率令牌存储桶中获取令牌。如果代币不可用，它将等待令牌变为可用。您SDK可能有配置选项可以让请求失败，而不必等待。存储桶中的令牌按照根据客户端收到的限制响应数量动态确定的速率进行充值。

`SendHTTPRequest`:

此步骤将请求发送到 AWS。 大多数 AWS SDKs在HTTP发出请求时，使用使用连接池重用现有连接的HTTP库。通常，如果请求由于限制错误而失败，则会重复使用连接，但如果请求由于暂时性错误而失败，则不会重复使用连接。

`RequestBookkeeping`:

如果请求成功，则会将令牌添加到令牌存储桶中。仅在`adaptive`重试模式下，请求速率令牌存储桶的填充率会根据收到的响应类型进行更新。

`Retryable`:

此步骤根据以下内容确定是否可以重试响应：

- HTTP状态码。
- 从服务返回的错误代码。
- 连接错误，定义为收到的任何未收到服务HTTP响应的错误。SDK

暂时错误（HTTP状态代码 400、408、500、502、503 和 504）和限制错误（HTTP状态代码 400、403、429、502、503 和 509）都可能被重试。SDK重试行为是结合错误代码或服务中的其他数据来确定的。

`MAX_ATTEMPTS`:

默认的最大尝试次数由设置`retry_mode`设置，除非被设置所`max_attempts`覆盖。

HasRetryQuota

此步骤从重试令牌存储桶中获取令牌。如果重试令牌存储桶为空，则不会重试请求。

ExponentialBackoff

对于可以重试的错误，使用截断的指数回退来计算重试延迟。SDKs使用带抖动的截断二进制指数回退。以下算法显示了如何为请求i的响应定义睡眠时间（以秒为单位）：

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

在上述算法中，以下值适用：

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

`r = 2`

`MAX_BACKOFF = 20 seconds`对于大多数人来说SDKs。请参阅您的特定SDK指南或源代码进行确认。

与之兼容 AWS SDKs

以下内容SDKs支持本主题中描述的功能和设置。所有部分例外情况均已注明。支持任何JVM系统属性设置 AWS SDK for Java 还有 AWS SDK for Kotlin 只有。

SDK	支持	备注或更多信息
AWS CLI v2	是	
SDK对于 C++	是	
SDK适用于 Go V2 (1.x)	是	
SDK适用于 Go 1.x (V1)	否	
SDK适用于 Java 2.x	是	
SDK适用于 Java 1.x	是	JVM系统属性：使用`com.amazonaws.sdk.maxAttempts`代替`aws.maxAttempts`；使用`com.amazonaws.sdk.retryMode`代替`aws.retryMode`。

SDK	支持	备注或更多信息
SDK适用于 JavaScript 3.x	是	
SDK适用于 JavaScript 2.x	否	支持最大重试次数、带抖动的指数回退以及用于重试回退的自定义方法选项。
SDK对于 Kotlin 来说	是	
SDK对于。 NET3.x	是	
SDK适用于 PHP 3.x	是	
SDK适用于 Python (Boto3)	是	
SDK适用于 Ruby 3.x	是	
SDK对于 Rust	是	
SDK为斯威夫特	是	
用于 PowerShell	是	

请求压缩

> **Note**
>
> 为了帮助理解设置页面的布局或解释 “兼容性” AWS SDKs下表，请参阅设置页面。

AWS SDKs而且工具可以在向发送请求时自动压缩有效负载 AWS 服务 支持接收压缩的有效载荷。在将有效负载发送到服务之前在客户端上对其进行压缩，可以减少向服务发送数据所需的请求总数和带宽，还可以减少由于服务对有效负载大小的限制而导致的失败请求。对于压缩，SDK或工具会选择服务和SDK。但是，当前可能的编码列表仅包含 gzip，但未来可能会扩展。

如果您的应用程序使用的是 Amazon，则请求压缩可能特别有用 CloudWatch。 CloudWatch 是一项监控和可观测性服务，它以日志、指标和事件的形式收集监控和操作数据。支持压缩的服务操作的一个示例 CloudWatch是PutMetricDataAPI的方法。

使用以下方法配置此功能：

disable_request_compression-共享 AWS **config**文件设置, **AWS_DISABLE_REQUEST_COMPRESSION** - 环境变量, **aws.disableRequestCompression**-JVM 系统属性：仅限 Java/Kotlin

开启或关闭SDK或工具是否在发送请求之前压缩有效负载。

默认值：`false`

有效值：

- **true** – 关闭请求压缩。
- **false** – 尽可能使用请求压缩。

request_min_compression_size_bytes-共享 AWS **config**文件设置, **AWS_REQUEST_MIN_COMPRESSION_SIZE_BYTES** - 环境变量, **aws.requestMinCompressionSizeBytes**-JVM 系统属性：仅限 Java/Kotlin

设置SDK或工具应压缩的请求正文的最小大小（以字节为单位）。压缩后，小型有效载荷可能会变得更长，因此，将会有一个下限，使执行压缩变得有意义。该值包含首尾，大于或等于该值的请求大小将被压缩。

默认值：10240 字节

有效值：介于 0 到 10485760 字节（包含首尾）之间的整数值。

与之兼容 AWS SDKs

以下内容SDKs支持本主题中描述的功能和设置。所有部分例外情况均已注明。支持任何JVM系统属性设置 AWS SDK for Java 还有 AWS SDK for Kotlin 只有。

SDK	支持	备注或更多信息
AWS CLI v2	是	
SDK对于 C++	是	
SDK适用于 Go V2 (1.x)	是	
SDK适用于 Go 1.x (V1)	否	

SDK	支持	备注或更多信息
SDK适用于 Java 2.x	是	
SDK适用于 Java 1.x	否	
SDK适用于 JavaScript 3.x	是	
SDK适用于 JavaScript 2.x	否	
SDK对于 Kotlin 来说	是	
SDK对于。 NET3.x	是	
SDK适用于 PHP 3.x	是	
SDK适用于 Python (Boto3)	是	
SDK适用于 Ruby 3.x	是	
SDK对于 Rust	是	
SDK为斯威夫特	否	
用于 PowerShell	是	

特定于服务的端点

服务特定的终端节点配置提供了使用您选择的终端节点进行API请求的选项，并使该选择保持不变。这些设置提供了支持本地端点、端点和第三方本地VPC端点的灵活性 AWS 开发环境。不同的端点可分别用于测试环境和生产环境。您可以URL为个人指定终端节点 AWS 服务.

使用以下方法配置此功能：

`endpoint_url`-共享 AWS **`config`**文件设置, **`AWS_ENDPOINT_URL`** - 环境变量, **`aws.endpointUrl`**-JVM 系统属性：仅限 Java/Kotlin

> 直接在配置文件中指定或作为环境变量指定时，此设置将指定用于所有服务请求的端点。此端点会被任何已配置的特定服务端点覆盖。

您也可以在共享的services部分中使用此设置 AWS config文件，用于为特定服务设置自定义终端节点。有关 services 节的子节中要使用的所有服务标识符密钥的列表，请参阅特定于服务的端点的标识符。

默认值：none

有效值：A URL 包括端点的方案和主机。URL可以选择包含包含一个或多个路径段的路径组件。

AWS_ENDPOINT_URL_<SERVICE> - 环境变量, **aws.endpointUrl<ServiceName>**-JVM 系统属性：仅限 Java/Kotlin

AWS_ENDPOINT_URL_<SERVICE>，在<SERVICE>哪里 AWS 服务 标识符，为特定服务设置自定义终端节点。有关特定于服务的所有环境变量的列表，请参阅特定于服务的端点的标识符。

此特定服务端点会覆盖 AWS_ENDPOINT_URL 中设置的任何全局端点。

默认值：none

有效值：A URL 包括端点的方案和主机。URL可以选择包含包含一个或多个路径段的路径组件。

ignore_configured_endpoint_urls-共享 AWS **config**文件设置, **AWS_IGNORE_CONFIGURED_ENDPOINT_URLS** - 环境变量, **aws.ignoreConfiguredEndpointUrls**-JVM 系统属性：仅限 Java/Kotlin

此设置用于忽略所有自定义端点配置。

请注意，无论此设置如何，都将使用代码中或服务客户端本身上设置的任何显式端点。例如，在--endpoint-url命令行参数中加入带有 AWS CLI 命令或将端点URL传递给客户端构造函数将始终生效。

默认值：false

有效值：

- **true**— SDK 或工具不会从共享config文件或用于设置端点的环境变量中读取任何自定义配置选项URL。
- **false**— SDK 或工具使用共享config文件或环境变量中用户提供的任何可用端点。

使用环境变量来配置端点

要将所有服务的请求路由到自定义终端节点URL，请设置AWS_ENDPOINT_URL全局环境变量。

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

路由特定请求的路由 AWS 服务 对于自定义终端节点URL，请使用AWS_ENDPOINT_URL_<SERVICE>环境变量。 Amazon DynamoDB 有一serviceId个DynamoDB。对于此服务，端点URL环境变量为AWS_ENDPOINT_URL_DYNAMODB。此端点优先于在 AWS_ENDPOINT_URL 中为此服务设置的全局端点。

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

再举一个例子， AWS Elastic Beanstalk 有一serviceId个Elastic Beanstalk。这些区域有：AWS 服务 标识符基于API模型的标识符，将所有空格serviceId替换为下划线，并将所有字母大写。为设置适用于此服务的端点，相应的环境变量为 AWS_ENDPOINT_URL_ELASTIC_BEANSTALK。有关特定于服务的所有环境变量的列表，请参阅特定于服务的端点的标识符。

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

使用共享 config 文件配置端点

在共享 config 文件中，endpoint_url 用于不同位置以实现不同的功能。

- endpoint_url 直接在 profile 中指定会使该端点成为全局端点。
- endpoint_url 嵌套在 services 部分中的服务标识符密钥下，使该端点仅适用于向该服务发出的请求。有关在共享 config 文件中定义 services 节的详细信息，请参阅 配置文件的格式。

以下示例使用services定义来配置用于 Amazon S3 的服务特定终端URL节点和用于所有其他服务的自定义全局终端节点：

```
[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
```

单个配置文件可以为多个服务配置端点。此示例说明如何为 Amazon S3 设置服务特URLs定的终端节点 AWS Elastic Beanstalk 在同一个个人资料中。 AWS Elastic Beanstalk 有

一serviceId个Elastic Beanstalk。这些区域有： AWS 服务 标识符基于API模型的标识符，将所有空格serviceId替换为下划线，并将所有字母小写。因此，服务标识符密钥变为 elastic_beanstalk 且已开始在线设置该服务 elastic_beanstalk = 。有关 services 节中要使用的所有服务标识符密钥的列表，请参阅特定于服务的端点的标识符。

```
[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
```

“服务配置”节可以在多个配置文件中使用。例如，两个配置文件在更改其他配置文件属性时可以使用相同的 services 定义：

```
[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
```

使用基于角色的凭证在配置文件中配置端点

如果您的配置文件具有通过用于IAM代入角色功能的source_profile参数配置的基于角色的凭证，则SDK仅使用指定配置文件的服务配置。它不使用关联有角色的配置文件。例如，使用以下共享 config 文件：

```
[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
```

如果您使用个人资料B并在代码中调用 AmazonEC2，则终端节点将解析为https://profile-b-ec2-endpoint.aws。如果您的代码向其他任何服务发出请求，则端点解析将不遵循任何自定义逻辑。该端点不会解析到配置文件 A 中定义的全局端点。要使全局端点对配置文件 B 生效，您需要直接在配置文件 B 中设置 endpoint_url。有关 source_profile 设置的更多信息，请参阅代入角色凭证提供者。

设置的优先级

该功能设置为可以同时使用，但每项服务只有一个值会优先使用。对于API拨打给定对象的电话 AWS 服务，则使用以下顺序来选择值：

1. 在代码中或服务客户端本身上设置的任何显式设置均优先于其他任何设置。
   - 对于 AWS CLI，这是--endpoint-url命令行参数提供的值。对于SDK，显式赋值可以采用您在实例化时设置的参数的形式 AWS 服务 客户端或配置对象。
2. 由特定于服务的环境变量提供的值，例如 AWS_ENDPOINT_URL_DYNAMODB。
3. AWS_ENDPOINT_URL 全局端点环境变量提供的值。
4. 该 endpoint_url 设置提供的值嵌套在共享 config 文件的 services 部分中的服务标识符密钥下。
5. 共享 config 文件的 profile 中直接指定的 endpoint_url 设置提供的值。
6. 相应端点的任何默认端点 URL AWS 服务 是最后使用的。

兼容 AWS SDKs

以下内容SDKs支持本主题中描述的功能和设置。所有部分例外情况均已注明。支持任何JVM系统属性设置 AWS SDK for Java 还有 AWS SDK for Kotlin 只有。

SDK	支持	备注或更多信息
AWS CLI v2	是	

SDK	支持	备注或更多信息
SDK对于 C++	否	
SDK适用于 Go V2 (1.x)	是	
SDK适用于 Go 1.x (V1)	否	
SDK适用于 Java 2.x	是	
SDK适用于 Java 1.x	否	
SDK适用于 JavaScript 3.x	是	
SDK适用于 JavaScript 2.x	否	
SDK对于 Kotlin 来说	是	
SDK对于。 NET3.x	是	
SDK适用于 PHP 3.x	是	
SDK适用于 Python (Boto3)	是	
SDK适用于 Ruby 3.x	是	
SDK对于 Rust	否	
SDK为斯威夫特	否	
用于 PowerShell	是	

特定于服务的端点的标识符

有关如何以及在何处使用下表中的标识符的信息，请参阅 特定于服务的端点。

serviceId	共享的服务标识密钥 AWS **co** fil	**AWS_ENDPOINT_URL_<SERVICE>** 环境变量
AccessAnalyzer	ac ly	AWS_ENDPOINT_URL_ACCESSANALYZER
Account	ac	AWS_ENDPOINT_URL_ACCOUNT
ACM	ac	AWS_ENDPOINT_URL_ACM
ACM PCA	ac	AWS_ENDPOINT_URL_ACM_PCA
Alexa For Business	a: _F	AWS_ENDPOINT_URL_ALEXA_FOR_BUSINESS
amp	ar	AWS_ENDPOINT_URL_AMP
Amplify	ar	AWS_ENDPOINT_URL_AMPLIFY
AmplifyBackend	ar cl	AWS_ENDPOINT_URL_AMPLIFYBACKEND
AmplifyUIBuilder	ar bu	AWS_ENDPOINT_URL_AMPLIFYUIBUILDER
API Gateway	ap ay	AWS_ENDPOINT_URL_API_GATEWAY

serviceId	共享的服务标识密钥 AWS **config** file	**AWS_ENDPOINT_URL_<SERVICE>** 环境变量
ApiGatewayManagementApi	api-gatewaymanagement	AWS_ENDPOINT_URL_APIGATEWAYMANAGEMENTAPI
ApiGatewayV2	apigatewayv2	AWS_ENDPOINT_URL_APIGATEWAYV2
AppConfig	appconfig	AWS_ENDPOINT_URL_APPCONFIG
AppConfigData	appconfigdata	AWS_ENDPOINT_URL_APPCONFIGDATA
AppFabric	appfabric	AWS_ENDPOINT_URL_APPFABRIC
Appflow	appflow	AWS_ENDPOINT_URL_APPFLOW
AppIntegrations	appintegrations	AWS_ENDPOINT_URL_APPINTEGRATIONS
Application Auto Scaling	application-autoscaling	AWS_ENDPOINT_URL_APPLICATION_AUTO_SCALING

serviceId	共享的服务标识密钥 AWS **co** fil	**AWS_ENDPOINT_URL_<SERVICE>** 环境变量
Application Insights	ap or ts	AWS_ENDPOINT_URL_APPLICATION_INSIGHTS
ApplicationCostProfiler	ap or f:	AWS_ENDPOINT_URL_APPLICATIONCOSTPROFILER
App Mesh	ap	AWS_ENDPOINT_URL_APP_MESH
AppRunner	ap	AWS_ENDPOINT_URL_APPRUNNER
AppStream	ap	AWS_ENDPOINT_URL_APPSTREAM
AppSync	ap	AWS_ENDPOINT_URL_APPSYNC
ARC Zonal Shift	a: _s	AWS_ENDPOINT_URL_ARC_ZONAL_SHIFT
Artifact	a:	AWS_ENDPOINT_URL_ARTIFACT
Athena	at	AWS_ENDPOINT_URL_ATHENA

serviceId	共享的服务标识密钥 AWS **config** file	**AWS_ENDPOINT_URL_<SERVICE>** 环境变量
AuditManager	auditmanager	AWS_ENDPOINT_URL_AUDITMANAGER
Auto Scaling	auto_scaling	AWS_ENDPOINT_URL_AUTO_SCALING
Auto Scaling Plans	auto_scaling_plans	AWS_ENDPOINT_URL_AUTO_SCALING_PLANS
b2bi	b2bi	AWS_ENDPOINT_URL_B2BI
Backup	backup	AWS_ENDPOINT_URL_BACKUP
Backup Gateway	backup_gateway	AWS_ENDPOINT_URL_BACKUP_GATEWAY
BackupStorage	backupstorage	AWS_ENDPOINT_URL_BACKUPSTORAGE
Batch	batch	AWS_ENDPOINT_URL_BATCH
BCM Data Exports	bcm_data_exports	AWS_ENDPOINT_URL_BCM_DATA_EXPORTS
Bedrock	bedrock	AWS_ENDPOINT_URL_BEDROCK

serviceId	共享的服务标识密钥 AWS **co** fil	**AWS_ENDPOINT_URL_<SERVICE>** 环境变量
Bedrock Agent	be ge	AWS_ENDPOINT_URL_BEDROCK_AGENT
Bedrock Agent Runtime	be ge ir	AWS_ENDPOINT_URL_BEDROCK_AGENT_RUNTIME
Bedrock Runtime	be ur	AWS_ENDPOINT_URL_BEDROCK_RUNTIME
billingconductor	bi no	AWS_ENDPOINT_URL_BILLINGCONDUCTOR
Braket	br	AWS_ENDPOINT_URL_BRAKET
Budgets	bu	AWS_ENDPOINT_URL_BUDGETS
Cost Explorer	ce ce	AWS_ENDPOINT_URL_COST_EXPLORER
chatbot	ch	AWS_ENDPOINT_URL_CHATBOT
Chime	ch	AWS_ENDPOINT_URL_CHIME

serviceId	共享的服务标识密钥 AWS **co** fil	**AWS_ENDPOINT_URL_<SERVICE>** 环境变量
Chime SDK Identity	ch _i	AWS_ENDPOINT_URL_CHIME_SDK_IDENTITY
Chime SDK Media Pipelines	ch _m pe	AWS_ENDPOINT_URL_CHIME_SDK_MEDIA_PIPELINES
Chime SDK Meetings	ch _m	AWS_ENDPOINT_URL_CHIME_SDK_MEETINGS
Chime SDK Messaging	ch _m g	AWS_ENDPOINT_URL_CHIME_SDK_MESSAGING
Chime SDK Voice	ch _v	AWS_ENDPOINT_URL_CHIME_SDK_VOICE
CleanRooms	cl s	AWS_ENDPOINT_URL_CLEANROOMS
CleanRoomsML	cl sn	AWS_ENDPOINT_URL_CLEANROOMSML

serviceId	共享的服务标识密钥 AWS **config** file	**AWS_ENDPOINT_URL_<SERVICE>** 环境变量
Cloud9	cl	AWS_ENDPOINT_URL_CLOUD9
CloudControl	cl rc	AWS_ENDPOINT_URL_CLOUDCONTROL
CloudDirectory	cl cl	AWS_ENDPOINT_URL_CLOUDDIRECTORY
CloudFormation	cl af	AWS_ENDPOINT_URL_CLOUDFORMATION
CloudFront	cl t	AWS_ENDPOINT_URL_CLOUDFRONT
CloudFront KeyValueStore	cl t_ es	AWS_ENDPOINT_URL_CLOUDFRONT_KEYVALUESTORE
CloudHSM	cl	AWS_ENDPOINT_URL_CLOUDHSM
CloudHSM V2	cl v2	AWS_ENDPOINT_URL_CLOUDHSM_V2
CloudSearch	cl ch	AWS_ENDPOINT_URL_CLOUDSEARCH

serviceId	共享的服务标识密钥 AWS **co** fil	**AWS_ENDPOINT_URL_<SERVICE>** 环境变量
CloudSearch Domain	cl ch	AWS_ENDPOINT_URL_CLOUDSEARCH_DOMAIN
CloudTrail	cl l	AWS_ENDPOINT_URL_CLOUDTRAIL
CloudTrail Data	cl l_	AWS_ENDPOINT_URL_CLOUDTRAIL_DATA
CloudWatch	cl h	AWS_ENDPOINT_URL_CLOUDWATCH
codeartifact	co ac	AWS_ENDPOINT_URL_CODEARTIFACT
CodeBuild	co	AWS_ENDPOINT_URL_CODEBUILD
CodeCatalyst	co ys	AWS_ENDPOINT_URL_CODECATALYST
CodeCommit	co t	AWS_ENDPOINT_URL_CODECOMMIT

serviceId	共享的服务标识密钥 AWS **co** fil	**AWS_ENDPOINT_URL_<SERVICE>** 环境变量
CodeDeploy	co y	AWS_ENDPOINT_URL_CODEDEPLOY
CodeGuru Reviewer	co re	AWS_ENDPOINT_URL_CODEGURU_REVIEWER
CodeGuru Security	co se	AWS_ENDPOINT_URL_CODEGURU_SECURITY
CodeGuruProfiler	co ro	AWS_ENDPOINT_URL_CODEGURUPROFILER
CodePipeline	co in	AWS_ENDPOINT_URL_CODEPIPELINE
CodeStar	co	AWS_ENDPOINT_URL_CODESTAR
CodeStar connections	co co ns	AWS_ENDPOINT_URL_CODESTAR_CONNECTIONS
codestar notifications	co no ic	AWS_ENDPOINT_URL_CODESTAR_NOTIFICATIONS

serviceId	共享的服务标识密钥 AWS **co** file	**AWS_ENDPOINT_URL_<SERVICE>** 环境变量
Cognito Identity	co de	AWS_ENDPOINT_URL_COGNITO_IDENTITY
Cognito Identity Provider	co de ro	AWS_ENDPOINT_URL_COGNITO_IDENTITY_PROVIDER
Cognito Sync	co yn	AWS_ENDPOINT_URL_COGNITO_SYNC
Comprehend	co d	AWS_ENDPOINT_URL_COMPREHEND
ComprehendMedical	co dr	AWS_ENDPOINT_URL_COMPREHENDMEDICAL
Compute Optimizer	co pt	AWS_ENDPOINT_URL_COMPUTE_OPTIMIZER
Config Service	co rv	AWS_ENDPOINT_URL_CONFIG_SERVICE
Connect	co	AWS_ENDPOINT_URL_CONNECT

serviceId	共享的服务标识密钥 AWS **co** fil	**AWS_ENDPOINT_URL_<SERVICE>** 环境变量
Connect Contact Lens	co on ns	AWS_ENDPOINT_URL_CONNECT_CONTACT_LENS
ConnectCampaigns	co mp	AWS_ENDPOINT_URL_CONNECTCAMPAIGNS
ConnectCases	co se	AWS_ENDPOINT_URL_CONNECTCASES
ConnectParticipant	co rt	AWS_ENDPOINT_URL_CONNECTPARTICIPANT
ControlTower	co we	AWS_ENDPOINT_URL_CONTROLTOWER
Cost Optimization Hub	co mi hu	AWS_ENDPOINT_URL_COST_OPTIMIZATION_HUB
Cost and Usage Report Service	co us oi ce	AWS_ENDPOINT_URL_COST_AND_USAGE_REPO RT_SERVICE

serviceId	共享的服务标识密钥 AWS **co** fil	**AWS_ENDPOINT_URL_<SERVICE>** 环境变量
Customer Profiles	cu p:	AWS_ENDPOINT_URL_CUSTOMER_PROFILES
DataBrew	da	AWS_ENDPOINT_URL_DATABREW
DataExchange	da ng	AWS_ENDPOINT_URL_DATAEXCHANGE
Data Pipeline	da l:	AWS_ENDPOINT_URL_DATA_PIPELINE
DataSync	da	AWS_ENDPOINT_URL_DATASYNC
DataZone	da	AWS_ENDPOINT_URL_DATAZONE
DAX	da	AWS_ENDPOINT_URL_DAX
Detective	de	AWS_ENDPOINT_URL_DETECTIVE
Device Farm	de rr	AWS_ENDPOINT_URL_DEVICE_FARM
DevOps Guru	de ru	AWS_ENDPOINT_URL_DEVOPS_GURU

serviceId	共享的服务标识密钥 AWS config file	AWS_ENDPOINT_URL_<SERVICE> 环境变量
Direct Connect	dirconnect	AWS_ENDPOINT_URL_DIRECT_CONNECT
Application Discovery Service	applicationdiscoveryservice	AWS_ENDPOINT_URL_APPLICATION_DISCOVERY_SERVICE
DLM	dlm	AWS_ENDPOINT_URL_DLM
Database Migration Service	databasemigration_service	AWS_ENDPOINT_URL_DATABASE_MIGRATION_SERVICE
DocDB	docdb	AWS_ENDPOINT_URL_DOCDB
DocDB Elastic	docdb-elastic	AWS_ENDPOINT_URL_DOCDB_ELASTIC
drs	drs	AWS_ENDPOINT_URL_DRS
Directory Service	directory_service	AWS_ENDPOINT_URL_DIRECTORY_SERVICE
DynamoDB	dynamodb	AWS_ENDPOINT_URL_DYNAMODB

serviceId	共享的服务标识密钥 AWS **config** file	**AWS_ENDPOINT_URL_<SERVICE>** 环境变量
DynamoDB Streams	dynamodb_streams	AWS_ENDPOINT_URL_DYNAMODB_STREAMS
EBS	ebs	AWS_ENDPOINT_URL_EBS
EC2	ec2	AWS_ENDPOINT_URL_EC2
EC2 Instance Connect	ec2_instance_connect	AWS_ENDPOINT_URL_EC2_INSTANCE_CONNECT
ECR	ecr	AWS_ENDPOINT_URL_ECR
ECR PUBLIC	ecr_public	AWS_ENDPOINT_URL_ECR_PUBLIC
ECS	ecs	AWS_ENDPOINT_URL_ECS
EFS	efs	AWS_ENDPOINT_URL_EFS
EKS	eks	AWS_ENDPOINT_URL_EKS
EKS Auth	eks_auth	AWS_ENDPOINT_URL_EKS_AUTH
Elastic Inference	elastic_inference	AWS_ENDPOINT_URL_ELASTIC_INFERENCE

serviceId	共享的服务标识密钥 AWS **co** fil	**AWS_ENDPOINT_URL_<SERVICE>** 环境变量
ElastiCache	el he	AWS_ENDPOINT_URL_ELASTICACHE
Elastic Beanstalk	el ea	AWS_ENDPOINT_URL_ELASTIC_BEANSTALK
Elastic Transcoder	el ra	AWS_ENDPOINT_URL_ELASTIC_TRANSCODER
Elastic Load Balancing	el oa c:	AWS_ENDPOINT_URL_ELASTIC_LOAD_BALANCING
Elastic Load Balancing v2	el oa c:	AWS_ENDPOINT_URL_ELASTIC_LOAD_BALANCING_V2
EMR	er	AWS_ENDPOINT_URL_EMR
EMR containers	er in	AWS_ENDPOINT_URL_EMR_CONTAINERS
EMR Serverless	er r:	AWS_ENDPOINT_URL_EMR_SERVERLESS

serviceId	共享的服务标识密钥 AWS **co** fil	**AWS_ENDPOINT_URL_<SERVICE>** 环境变量
EntityResolution	er o:	AWS_ENDPOINT_URL_ENTITYRESOLUTION
Elasticsearch Service	e: a: i	AWS_ENDPOINT_URL_ELASTICSEARCH_SERVICE
EventBridge	ev ge	AWS_ENDPOINT_URL_EVENTBRIDGE
Evidently	ev	AWS_ENDPOINT_URL_EVIDENTLY
finspace	f:	AWS_ENDPOINT_URL_FINSPACE
finspace data	f: da	AWS_ENDPOINT_URL_FINSPACE_DATA
Firehose	f:	AWS_ENDPOINT_URL_FIREHOSE
fis	f:	AWS_ENDPOINT_URL_FIS
FMS	fr	AWS_ENDPOINT_URL_FMS
forecast	fc	AWS_ENDPOINT_URL_FORECAST

serviceId	共享的服务标识密钥 AWS **co** fil	**AWS_ENDPOINT_URL_<SERVICE>** 环境变量
forecastquery	fo ue	AWS_ENDPOINT_URL_FORECASTQUERY
FraudDetector	fr ct	AWS_ENDPOINT_URL_FRAUDDETECTOR
FreeTier	fr	AWS_ENDPOINT_URL_FREETIER
FSx	fs	AWS_ENDPOINT_URL_FSX
GameLift	ga	AWS_ENDPOINT_URL_GAMELIFT
Glacier	gl	AWS_ENDPOINT_URL_GLACIER
Global Accelerator	gl ce	AWS_ENDPOINT_URL_GLOBAL_ACCELERATOR
Glue	gl	AWS_ENDPOINT_URL_GLUE
grafana	gr	AWS_ENDPOINT_URL_GRAFANA
Greengrass	gr s	AWS_ENDPOINT_URL_GREENGRASS

serviceId	共享的服务标识密钥 AWS **co** fil	**AWS_ENDPOINT_URL_<SERVICE>** 环境变量
GreengrassV2	gr sv	AWS_ENDPOINT_URL_GREENGRASSV2
GroundStation	gr st	AWS_ENDPOINT_URL_GROUNDSTATION
GuardDuty	gu	AWS_ENDPOINT_URL_GUARDDUTY
Health	he	AWS_ENDPOINT_URL_HEALTH
HealthLake	he e	AWS_ENDPOINT_URL_HEALTHLAKE
Honeycode	ho	AWS_ENDPOINT_URL_HONEYCODE
IAM	ia	AWS_ENDPOINT_URL_IAM
identitystore	id to	AWS_ENDPOINT_URL_IDENTITYSTORE
imagebuilder	im de	AWS_ENDPOINT_URL_IMAGEBUILDER

serviceId	共享的服务标识密钥 AWS **co** fil	**AWS_ENDPOINT_URL_<SERVICE>** 环境变量
ImportExport	ir o:	AWS_ENDPOINT_URL_IMPORTEXPORT
Inspector	ir	AWS_ENDPOINT_URL_INSPECTOR
Inspector Scan	ir _s	AWS_ENDPOINT_URL_INSPECTOR_SCAN
Inspector2	ir 2	AWS_ENDPOINT_URL_INSPECTOR2
InternetMonitor	ir or	AWS_ENDPOINT_URL_INTERNETMONITOR
IoT	io	AWS_ENDPOINT_URL_IOT
IoT Data Plane	io p:	AWS_ENDPOINT_URL_IOT_DATA_PLANE
IoT Jobs Data Plane	io da e	AWS_ENDPOINT_URL_IOT_JOBS_DATA_PLANE

serviceId	共享的服务标识密钥 AWS **co** fil	**AWS_ENDPOINT_URL_<SERVICE>** 环境变量
IoT 1Click Devices Service	ic k_ _S	AWS_ENDPOINT_URL_IOT_1CLICK_DEVICES_ SERVICE
IoT 1Click Projects	ic k_ s	AWS_ENDPOINT_URL_IOT_1CLICK_PROJECTS
IoTAnalytics	ic ic	AWS_ENDPOINT_URL_IOTANALYTICS
IotDeviceAdvisor	ic ad	AWS_ENDPOINT_URL_IOTDEVICEADVISOR
IoT Events	ic s	AWS_ENDPOINT_URL_IOT_EVENTS
IoT Events Data	ic s_	AWS_ENDPOINT_URL_IOT_EVENTS_DATA
IoTFleetHub	ic ub	AWS_ENDPOINT_URL_IOTFLEETHUB
IoTFleetWise	ic is	AWS_ENDPOINT_URL_IOTFLEETWISE

serviceId	共享的服务标识密钥 AWS **co** fil	**AWS_ENDPOINT_URL_<SERVICE>** 环境变量
IoTSecureTunneling	io tu	AWS_ENDPOINT_URL_IOTSECURETUNNELING
IoTSiteWise	io se	AWS_ENDPOINT_URL_IOTSITEWISE
IoTThingsGraph	io gr	AWS_ENDPOINT_URL_IOTTHINGSGRAPH
IoTTwinMaker	io ke	AWS_ENDPOINT_URL_IOTTWINMAKER
IoT Wireless	io es	AWS_ENDPOINT_URL_IOT_WIRELESS
ivs	iv	AWS_ENDPOINT_URL_IVS
IVS RealTime	iv ir	AWS_ENDPOINT_URL_IVS_REALTIME
ivschat	iv	AWS_ENDPOINT_URL_IVSCHAT
Kafka	ka	AWS_ENDPOINT_URL_KAFKA

serviceId	共享的服务标识密钥 AWS **co** fil	**AWS_ENDPOINT_URL_<SERVICE>** 环境变量
KafkaConnect	ka ec	AWS_ENDPOINT_URL_KAFKACONNECT
kendra	ke	AWS_ENDPOINT_URL_KENDRA
Kendra Ranking	ke nl	AWS_ENDPOINT_URL_KENDRA_RANKING
Keyspaces	ke	AWS_ENDPOINT_URL_KEYSPACES
Kinesis	ki	AWS_ENDPOINT_URL_KINESIS
Kinesis Video Archived Media	ki id iv a	AWS_ENDPOINT_URL_KINESIS_VIDEO_ARCHIVED_MEDIA
Kinesis Video Media	ki id a	AWS_ENDPOINT_URL_KINESIS_VIDEO_MEDIA
Kinesis Video Signaling	ki id a	AWS_ENDPOINT_URL_KINESIS_VIDEO_SIGNALING

serviceId	共享的服务标识密钥 AWS **config** file	**AWS_ENDPOINT_URL_<SERVICE>** 环境变量
Kinesis Video WebRTC Storage	kinesisvideowebrtcstorage	AWS_ENDPOINT_URL_KINESIS_VIDEO_WEBRTC_STORAGE
Kinesis Analytics	kinesisanalytics	AWS_ENDPOINT_URL_KINESIS_ANALYTICS
Kinesis Analytics V2	kinesisanalyticsv2	AWS_ENDPOINT_URL_KINESIS_ANALYTICS_V2
Kinesis Video	kinesisvideo	AWS_ENDPOINT_URL_KINESIS_VIDEO
KMS	kms	AWS_ENDPOINT_URL_KMS
LakeFormation	lakeformation	AWS_ENDPOINT_URL_LAKEFORMATION
Lambda	lambda	AWS_ENDPOINT_URL_LAMBDA
Launch Wizard	launchwizard	AWS_ENDPOINT_URL_LAUNCH_WIZARD

serviceId	共享的服务标识密钥 AWS **co** fil	**AWS_ENDPOINT_URL_<SERVICE>** 环境变量
Lex Model Building Service	le _l _s	AWS_ENDPOINT_URL_LEX_MODEL_BUILDING_ SERVICE
Lex Runtime Service	le me e	AWS_ENDPOINT_URL_LEX_RUNTIME_SERVICE
Lex Models V2	le s_	AWS_ENDPOINT_URL_LEX_MODELS_V2
Lex Runtime V2	le me	AWS_ENDPOINT_URL_LEX_RUNTIME_V2
License Manager	li ar	AWS_ENDPOINT_URL_LICENSE_MANAGER
License Manager Linux Subscriptions	li ar nu r:	AWS_ENDPOINT_URL_LICENSE_MANAGER_LIN UX_SUBSCRIPTIONS

serviceId	共享的服务标识密钥 AWS **co** fil	**AWS_ENDPOINT_URL_<SERVICE>** 环境变量
License Manager User Subscriptions	l: ar e: ip	AWS_ENDPOINT_URL_LICENSE_MANAGER_USER_SUBSCRIPTIONS
Lightsail	l:	AWS_ENDPOINT_URL_LIGHTSAIL
Location	l(	AWS_ENDPOINT_URL_LOCATION
CloudWatch Logs	c: h_	AWS_ENDPOINT_URL_CLOUDWATCH_LOGS
CloudWatch Logs	c: h_	AWS_ENDPOINT_URL_CLOUDWATCH_LOGS
LookoutEquipment	l(u:	AWS_ENDPOINT_URL_LOOKOUTEQUIPMENT
LookoutMetrics	l(t:	AWS_ENDPOINT_URL_LOOKOUTMETRICS
LookoutVision	l(s:	AWS_ENDPOINT_URL_LOOKOUTVISION

serviceId	共享的服务标识密钥 AWS **config** file	**AWS_ENDPOINT_URL_<SERVICE>** 环境变量
m2	m2	AWS_ENDPOINT_URL_M2
Machine Learning	machinelearning	AWS_ENDPOINT_URL_MACHINE_LEARNING
Macie2	macie2	AWS_ENDPOINT_URL_MACIE2
ManagedBlockchain	managedblockchain	AWS_ENDPOINT_URL_MANAGEDBLOCKCHAIN
ManagedBlockchain Query	managedblockchain-query	AWS_ENDPOINT_URL_MANAGEDBLOCKCHAIN_QUERY
Marketplace Agreement	marketplace-agreement	AWS_ENDPOINT_URL_MARKETPLACE_AGREEMENT
Marketplace Catalog	marketplace-catalog	AWS_ENDPOINT_URL_MARKETPLACE_CATALOG
Marketplace Deployment	marketplace-deployment	AWS_ENDPOINT_URL_MARKETPLACE_DEPLOYMENT

serviceId	共享的服务标识密钥 AWS **config** file	**AWS_ENDPOINT_URL_<SERVICE>** 环境变量
Marketplace Entitlement Service	marketplace-entitlement-service	AWS_ENDPOINT_URL_MARKETPLACE_ENTITLEMENT_SERVICE
Marketplace Commerce Analytics	marketplacecommerceanalytics	AWS_ENDPOINT_URL_MARKETPLACE_COMMERCE_ANALYTICS
MediaConnect	mediaconnect	AWS_ENDPOINT_URL_MEDIACONNECT
MediaConvert	mediaconvert	AWS_ENDPOINT_URL_MEDIACONVERT
MediaLive	medialive	AWS_ENDPOINT_URL_MEDIALIVE
MediaPackage	mediapackage	AWS_ENDPOINT_URL_MEDIAPACKAGE
MediaPackage Vod	mediapackage-vod	AWS_ENDPOINT_URL_MEDIAPACKAGE_VOD

serviceId	共享的服务标识密钥 AWS **co** fil	**AWS_ENDPOINT_URL_<SERVICE>** 环境变量
MediaPackageV2	me ag	AWS_ENDPOINT_URL_MEDIAPACKAGEV2
MediaStore	me e	AWS_ENDPOINT_URL_MEDIASTORE
MediaStore Data	me e_	AWS_ENDPOINT_URL_MEDIASTORE_DATA
MediaTailor	me or	AWS_ENDPOINT_URL_MEDIATAILOR
Medical Imaging	me ma	AWS_ENDPOINT_URL_MEDICAL_IMAGING
MemoryDB	me	AWS_ENDPOINT_URL_MEMORYDB
Marketplace Metering	ma ce ng	AWS_ENDPOINT_URL_MARKETPLACE_METERING
Migration Hub	mi _h	AWS_ENDPOINT_URL_MIGRATION_HUB
mgn	mg	AWS_ENDPOINT_URL_MGN

serviceId	共享的服务标识密钥 AWS **co** fil	**AWS_ENDPOINT_URL_<SERVICE>** 环境变量
Migration Hub Refactor Spaces	mi _l ct es	AWS_ENDPOINT_URL_MIGRATION_HUB_REFAC TOR_SPACES
MigrationHub Config	mi hu g	AWS_ENDPOINT_URL_MIGRATIONHUB_CONFIG
MigrationHubOrches trator	mi hu t:	AWS_ENDPOINT_URL_MIGRATIONHUBORCHESTRATOR
MigrationHubStrategy	mi hu gy	AWS_ENDPOINT_URL_MIGRATIONHUBSTRATEGY
Mobile	mo	AWS_ENDPOINT_URL_MOBILE
mq	mo	AWS_ENDPOINT_URL_MQ
MTurk	mt	AWS_ENDPOINT_URL_MTURK
MWAA	mw	AWS_ENDPOINT_URL_MWAA

serviceId	共享的服务标识密钥 AWS config file	AWS_ENDPOINT_URL_<SERVICE> 环境变量
Neptune	ne	AWS_ENDPOINT_URL_NEPTUNE
Neptune Graph	ne ra	AWS_ENDPOINT_URL_NEPTUNE_GRAPH
neptunedata	ne ta	AWS_ENDPOINT_URL_NEPTUNEDATA
Network Firewall	ne i:	AWS_ENDPOINT_URL_NETWORK_FIREWALL
NetworkManager	ne na	AWS_ENDPOINT_URL_NETWORKMANAGER
NetworkMonitor	ne n:	AWS_ENDPOINT_URL_NETWORKMONITOR
nimble	n:	AWS_ENDPOINT_URL_NIMBLE
OAM	oa	AWS_ENDPOINT_URL_OAM
Omics	or	AWS_ENDPOINT_URL_OMICS
OpenSearch	op h	AWS_ENDPOINT_URL_OPENSEARCH

serviceId	共享的服务标识密钥 AWS config file	AWS_ENDPOINT_URL_<SERVICE> 环境变量
OpenSearchServerless	op hs ss	AWS_ENDPOINT_URL_OPENSEARCHSERVERLESS
OpsWorks	op	AWS_ENDPOINT_URL_OPSWORKS
OpsWorksCM	op m	AWS_ENDPOINT_URL_OPSWORKSCM
Organizations	or id	AWS_ENDPOINT_URL_ORGANIZATIONS
OSIS	os	AWS_ENDPOINT_URL_OSIS
Outposts	ou	AWS_ENDPOINT_URL_OUTPOSTS
p8data	p8	AWS_ENDPOINT_URL_P8DATA
p8data	p8	AWS_ENDPOINT_URL_P8DATA
Panorama	pa	AWS_ENDPOINT_URL_PANORAMA
Payment Cryptography	pa ry hy	AWS_ENDPOINT_URL_PAYMENT_CRYPTOGRAPHY

serviceId	共享的服务标识密钥 AWS **co** fil	**AWS_ENDPOINT_URL_<SERVICE>** 环境变量
Payment Cryptography Data	pa ry hy	AWS_ENDPOINT_URL_PAYMENT_CRYPTOGRAPHY_DATA
Pca Connector Ad	pc ct	AWS_ENDPOINT_URL_PCA_CONNECTOR_AD
Personalize	pe ze	AWS_ENDPOINT_URL_PERSONALIZE
Personalize Events	pe ze	AWS_ENDPOINT_URL_PERSONALIZE_EVENTS
Personalize Runtime	pe ze e	AWS_ENDPOINT_URL_PERSONALIZE_RUNTIME
PI	pi	AWS_ENDPOINT_URL_PI
Pinpoint	pi	AWS_ENDPOINT_URL_PINPOINT
Pinpoint Email	pi er	AWS_ENDPOINT_URL_PINPOINT_EMAIL

serviceId	共享的服务标识密钥 AWS **co** fil	**AWS_ENDPOINT_URL_<SERVICE>** 环境变量
Pinpoint SMS Voice	p: sr	AWS_ENDPOINT_URL_PINPOINT_SMS_VOICE
Pinpoint SMS Voice V2	p: sr _\	AWS_ENDPOINT_URL_PINPOINT_SMS_VOICE_V2
Pipes	p:	AWS_ENDPOINT_URL_PIPES
Polly	p(	AWS_ENDPOINT_URL_POLLY
Pricing	p:	AWS_ENDPOINT_URL_PRICING
PrivateNetworks	p: tv	AWS_ENDPOINT_URL_PRIVATENETWORKS
Proton	p:	AWS_ENDPOINT_URL_PROTON
QBusiness	ql	AWS_ENDPOINT_URL_QBUSINESS
QConnect	q(	AWS_ENDPOINT_URL_QCONNECT
QLDB	q:	AWS_ENDPOINT_URL_QLDB

serviceId	共享的服务标识密钥 AWS **co** fil	**AWS_ENDPOINT_URL_<SERVICE>** 环境变量
QLDB Session	q ic	AWS_ENDPOINT_URL_QLDB_SESSION
QuickSight	qu t	AWS_ENDPOINT_URL_QUICKSIGHT
RAM	ra	AWS_ENDPOINT_URL_RAM
rbin	rb	AWS_ENDPOINT_URL_RBIN
RDS	rd	AWS_ENDPOINT_URL_RDS
RDS Data	rd	AWS_ENDPOINT_URL_RDS_DATA
Redshift	re	AWS_ENDPOINT_URL_REDSHIFT
Redshift Data	re da	AWS_ENDPOINT_URL_REDSHIFT_DATA
Redshift Serverless	re se s	AWS_ENDPOINT_URL_REDSHIFT_SERVERLESS
Rekognition	re or	AWS_ENDPOINT_URL_REKOGNITION

serviceId	共享的服务标识密钥 AWS **co** fil	**AWS_ENDPOINT_URL_<SERVICE>** 环境变量
repostspace	re ce	AWS_ENDPOINT_URL_REPOSTSPACE
resiliencehub	re eh	AWS_ENDPOINT_URL_RESILIENCEHUB
Resource Explorer 2	re ex 2	AWS_ENDPOINT_URL_RESOURCE_EXPLORER_2
Resource Groups	re gr	AWS_ENDPOINT_URL_RESOURCE_GROUPS
Resource Groups Tagging API	re gr gg	AWS_ENDPOINT_URL_RESOURCE_GROUPS_TAG GING_API
RoboMaker	ro	AWS_ENDPOINT_URL_ROBOMAKER
RolesAnywhere	ro he	AWS_ENDPOINT_URL_ROLESANYWHERE
Route 53	ro	AWS_ENDPOINT_URL_ROUTE_53

serviceId	共享的服务标识密钥 AWS **co** fil	**AWS_ENDPOINT_URL_<SERVICE>** 环境变量
Route53 Recovery Cluster	ro ec lu	AWS_ENDPOINT_URL_ROUTE53_RECOVERY_CLUSTER
Route53 Recovery Control Config	ro ec on nt	AWS_ENDPOINT_URL_ROUTE53_RECOVERY_CONTROL_CONFIG
Route53 Recovery Readiness	ro ec ea	AWS_ENDPOINT_URL_ROUTE53_RECOVERY_READINESS
Route 53 Domains	ro do	AWS_ENDPOINT_URL_ROUTE_53_DOMAINS
Route53Resolver	ro so	AWS_ENDPOINT_URL_ROUTE53RESOLVER
RUM	ru	AWS_ENDPOINT_URL_RUM
S3	s3	AWS_ENDPOINT_URL_S3
S3 Control	s3 l	AWS_ENDPOINT_URL_S3_CONTROL

serviceId	共享的服务标识密钥 AWS **co** fil	**AWS_ENDPOINT_URL_<SERVICE>** 环境变量
S3Outposts	s3 s	AWS_ENDPOINT_URL_S3OUTPOSTS
SageMaker	sa	AWS_ENDPOINT_URL_SAGEMAKER
SageMaker A2I Runtime	sa _a in	AWS_ENDPOINT_URL_SAGEMAKER_A2I_RUNTIME
Sagemaker Edge	sa _e	AWS_ENDPOINT_URL_SAGEMAKER_EDGE
SageMaker FeatureStore Runtime	sa _f to in	AWS_ENDPOINT_URL_SAGEMAKER_FEATUREST ORE_RUNTIME
SageMaker Geospatial	sa _g a1	AWS_ENDPOINT_URL_SAGEMAKER_GEOSPATIAL
SageMaker Metrics	sa _m	AWS_ENDPOINT_URL_SAGEMAKER_METRICS

serviceId	共享的服务标识密钥 AWS **co** fil	**AWS_ENDPOINT_URL_<SERVICE>** 环境变量
SageMaker Runtime	sa _	AWS_ENDPOINT_URL_SAGEMAKER_RUNTIME
savingsplans	sa ar	AWS_ENDPOINT_URL_SAVINGSPLANS
Scheduler	sc	AWS_ENDPOINT_URL_SCHEDULER
schemas	sc	AWS_ENDPOINT_URL_SCHEMAS
SimpleDB	sd	AWS_ENDPOINT_URL_SIMPLEDB
Secrets Manager	se ar	AWS_ENDPOINT_URL_SECRETS_MANAGER
SecurityHub	se ub	AWS_ENDPOINT_URL_SECURITYHUB
SecurityLake	se al	AWS_ENDPOINT_URL_SECURITYLAKE

serviceId	共享的服务标识密钥 AWS **co** fil	**AWS_ENDPOINT_URL_<SERVICE>** 环境变量
ServerlessApplicationRepository	se sa io to	AWS_ENDPOINT_URL_SERVERLESSAPPLICATIONREPOSITORY
Service Quotas	se uo	AWS_ENDPOINT_URL_SERVICE_QUOTAS
Service Catalog	se at	AWS_ENDPOINT_URL_SERVICE_CATALOG
Service Catalog AppRegistry	se at pi	AWS_ENDPOINT_URL_SERVICE_CATALOG_APPREGISTRY
ServiceDiscovery	se sc	AWS_ENDPOINT_URL_SERVICEDISCOVERY
SES	se	AWS_ENDPOINT_URL_SES
SESv2	se	AWS_ENDPOINT_URL_SESV2
Shield	sh	AWS_ENDPOINT_URL_SHIELD

serviceId	共享的服务标识密钥 AWS **config** file	**AWS_ENDPOINT_URL_<SERVICE>** 环境变量
signer	signer	AWS_ENDPOINT_URL_SIGNER
SimSpaceWeaver	simspaceweaver	AWS_ENDPOINT_URL_SIMSPACEWEAVER
SMS	sms	AWS_ENDPOINT_URL_SMS
Snow Device Management	snow-device-management	AWS_ENDPOINT_URL_SNOW_DEVICE_MANAGEMENT
Snowball	snowball	AWS_ENDPOINT_URL_SNOWBALL
SNS	sns	AWS_ENDPOINT_URL_SNS
SQS	sqs	AWS_ENDPOINT_URL_SQS
SSM	ssm	AWS_ENDPOINT_URL_SSM
SSM Contacts	ssm-contacts	AWS_ENDPOINT_URL_SSM_CONTACTS
SSM Incidents	ssm-incidents	AWS_ENDPOINT_URL_SSM_INCIDENTS
Ssm Sap	ssm-sap	AWS_ENDPOINT_URL_SSM_SAP

serviceId	共享的服务标识密钥 AWS **co** fil	**AWS_ENDPOINT_URL_<SERVICE>** 环境变量
SSO	ss	AWS_ENDPOINT_URL_SSO
SSO Admin	ss	AWS_ENDPOINT_URL_SSO_ADMIN
SSO OIDC	ss	AWS_ENDPOINT_URL_SSO_OIDC
SFN	st	AWS_ENDPOINT_URL_SFN
Storage Gateway	st at	AWS_ENDPOINT_URL_STORAGE_GATEWAY
STS	st	AWS_ENDPOINT_URL_STS
SupplyChain	su in	AWS_ENDPOINT_URL_SUPPLYCHAIN
Support	su	AWS_ENDPOINT_URL_SUPPORT
Support App	su pp	AWS_ENDPOINT_URL_SUPPORT_APP
SWF	sw	AWS_ENDPOINT_URL_SWF
synthetics	sy s	AWS_ENDPOINT_URL_SYNTHETICS

serviceId	共享的服务标识密钥 AWS **co** fil	**AWS_ENDPOINT_URL_<SERVICE>** 环境变量
Textract	te	AWS_ENDPOINT_URL_TEXTRACT
Timestream InfluxDB	ti m_ b	AWS_ENDPOINT_URL_TIMESTREAM_INFLUXDB
Timestream Query	ti m_	AWS_ENDPOINT_URL_TIMESTREAM_QUERY
Timestream Write	ti m_	AWS_ENDPOINT_URL_TIMESTREAM_WRITE
tnb	tr	AWS_ENDPOINT_URL_TNB
Transcribe	ti e	AWS_ENDPOINT_URL_TRANSCRIBE
Transfer	ti	AWS_ENDPOINT_URL_TRANSFER
Translate	ti	AWS_ENDPOINT_URL_TRANSLATE
TrustedAdvisor	ti v:	AWS_ENDPOINT_URL_TRUSTEDADVISOR

serviceId	共享的服务标识密钥 AWS_ce file	AWS_ENDPOINT_URL_<SERVICE> 环境变量
VerifiedPermissions	ve es	AWS_ENDPOINT_URL_VERIFIEDPERMISSIONS
Voice ID	vo	AWS_ENDPOINT_URL_VOICE_ID
VPC Lattice	vp ce	AWS_ENDPOINT_URL_VPC_LATTICE
WAF	wa	AWS_ENDPOINT_URL_WAF
WAF Regional	wa na	AWS_ENDPOINT_URL_WAF_REGIONAL
WAFV2	wa	AWS_ENDPOINT_URL_WAFV2
WellArchitected	we te	AWS_ENDPOINT_URL_WELLARCHITECTED
Wisdom	wi	AWS_ENDPOINT_URL_WISDOM
WorkDocs	wo	AWS_ENDPOINT_URL_WORKDOCS
WorkLink	wo	AWS_ENDPOINT_URL_WORKLINK
WorkMail	wo	AWS_ENDPOINT_URL_WORKMAIL

serviceId	共享的服务标识密钥 AWS **co** fil	**AWS_ENDPOINT_URL_<SERVICE>** 环境变量
WorkMailMessageFlow	wo es w	AWS_ENDPOINT_URL_WORKMAILMESSAGEFLOW
WorkSpaces	wo s	AWS_ENDPOINT_URL_WORKSPACES
WorkSpaces Thin Client	wo s_ ie	AWS_ENDPOINT_URL_WORKSPACES_THIN_CLIENT
WorkSpaces Web	wo s_	AWS_ENDPOINT_URL_WORKSPACES_WEB
XRay	x:	AWS_ENDPOINT_URL_XRAY

智能配置默认值

借助智能配置默认功能，AWS SDKs可以为其他配置设置提供预定义的、经过优化的默认值。

使用以下方法配置此功能：

defaults_mode-共享 AWS **config**文件设置, **AWS_DEFAULTS_MODE** - 环境变量, **aws.defaultsMode**-JVM 系统属性：仅限 Java/Kotlin

使用此设置，您可以选择与您的应用程序架构相匹配的模式，然后为您的应用程序提供经过优化的默认值。如果 AWS SDKsetting 已明确设置一个值，则该值始终优先。如果 AWS SDKsetting 没有明确设置值，也不等于旧版，则此功能可以为针对您的应用程序优化的各种设置提供不同的默认值。defaults_mode设置可能包括以下内容：HTTP通信设置、重试行为、服务区域终端节点设置，可能还包括任何SDK相关配置。使用此功能的客户可以获得针对常见使用场景量身定制的新配置默认值。如果不等于legacy，我们建议您在升级时对您的应用程序进行测试SDK，因为提供的默认值可能会随着最佳实践的发展而改变。defaults_mode

默认值：legacy

注意：will 的新主要SDKs版本默认为standard。

有效值：

- legacy— 提供默认设置，这些设置因而异，SDK并且在建立之前就已存在defaults_mode。
- standard – 提供最新的推荐默认值，这些默认值在大多数情况下都应该可以安全运行。
- in-region— 基于标准模式构建，包括为调用的应用程序量身定制的优化 AWS 服务 从同一个地方 AWS 区域.
- cross-region— 基于标准模式构建，包括为调用的应用程序量身定制的优化 AWS 服务 在不同的区域。
- mobile – 基于标准模式构建，包括为移动应用程序量身定制的优化。
- auto – 基于标准模式构建，包括实验功能。SDK尝试发现运行时环境以自动确定适当的设置。自动检测是基于启发式的，无法提供 100% 的准确性。如果无法确定运行时系统环境，则使用 standard 模式。auto 检测可能会查询实例元数据，这可能会带来延迟。如果启动延迟对您的应用程序而言至关重要，我们建议您改为选择显式 defaults_mode 延迟。

在 config 文件中设置此值的示例：

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

以下参数可能会根据 defaults_mode 的选项进行优化：

- retryMode— 指定SDK尝试的重试方式。请参阅 重试行为。
- stsRegionalEndpoints— 指定如何SDK确定 AWS 服务 它用来与之交谈的端点 AWS Security Token Service (AWS STS)。 见AWS STS 区域终端节点。

- s3UsEast1RegionalEndpoints— 指定如何SDK确定 AWS 用于与该us-east-1地区的 Amazon S3 通信的服务终端节点。
- connectTimeoutInMillis – 在套接字上进行初始连接尝试后，超时之前的时长。如果客户端没有收到连接握手完成的消息，则客户端会放弃操作并使其失败。
- tlsNegotiationTimeoutInMillis— 从发送CLIENTHELLO消息到客户端和服务器完全协商密码并交换密钥为止，TLS握手所花费的最大时间。

每个设置的默认值会根据为应用程序选择的 defaults_mode 而变化。这些值目前设置如下（可能会发生变化）：

参数	**standard** 模式	**in-region** 模式	**cross-region** 模式	**mobile** 模式
retryMode	standard	standard	standard	standard
stsRegionalEndpoints	regional	regional	regional	regional
s3UsEast1RegionalEndpoints	regional	regional	regional	regional
connectTimeoutInMillis	3100	1100	3100	30000
tlsNegotiationTimeoutInMillis	3100	1100	3100	30000

例如，如果您选择的defaults_mode是standard，则将为retry_mode分配standard的值（来自有效的retry_mode选项），将为stsRegionalEndpoints分配regional的值（来自有效stsRegionalEndpoints选项）。

兼容 AWS SDKs

以下内容SDKs支持本主题中描述的功能和设置。所有部分例外情况均已注明。支持任何JVM系统属性设置 AWS SDK for Java 还有 AWS SDK for Kotlin 只有。

SDK	支持	备注或更多信息
AWS CLI v2	否	
SDK对于 C++	是	参数未优化：stsRegionalEndpoints 、s3UsEast1RegionalEndpoints 、tlsNegotiationTimeoutInMillis 。
SDK适用于 Go V2 (1.x)	是	参数未优化：retryMode 、stsRegionalEndpoints 、s3UsEast1RegionalEndpoints 。
SDK适用于 Go 1.x (V1)	否	
SDK适用于 Java 2.x	是	参数未优化：stsRegionalEndpoints 。
SDK适用于 Java 1.x	否	
SDK适用于 JavaScript 3.x	是	参数未优化：stsRegionalEndpoints 、s3UsEast1RegionalEndpoints 、tlsNegotiationTimeoutInMillis 。connectTimeoutInMillis 被称为 connectionTimeout 。

SDK	支持	备注或更多信息
SDK适用于 JavaScript 2.x	否	
SDK对于 Kotlin 来说	否	
SDK对于。 NET3.x	是	参数未优化：connectTimeoutInMillis 、tlsNegotiationTimeoutInMillis 。
SDK适用于 PHP 3.x	是	参数未优化：tlsNegotiationTimeoutInMillis 。
SDK适用于 Python (Boto3)	是	参数未优化：tlsNegotiationTimeoutInMillis 。
SDK适用于 Ruby 3.x	是	
SDK对于 Rust	否	
SDK为斯威夫特	否	
用于 PowerShell	是	参数未优化：connectTimeoutInMillis 、tlsNegotiationTimeoutInMillis 。

AWS 常用运行时 (CRT) 库

C AWS ommon Runtime (CRT) 库是的基础库SDKs。CRT这是一个由独立封装组成的模块化系列，用 C 语言编写。每个封装都为不同的所需功能提供了良好的性能和最小的占用空间。这些功能是通用的，并且在所有功能之间共享，SDKs从而提供了更好的代码重用、优化和准确性。程序包是：

- awslabs/aws-c-auth: AWS 客户端身份验证（标准凭据提供程序和签名 (sigv4)）
- awslabs/aws-c-cal: 加密原始类型、哈希 (、、SHA256HMAC) MD5、签名SHA256者、AES
- awslabs/aws-c-common：基本数据结构、线程/同步原始类型、缓冲区管理、stdlib 相关函数
- awslabs/aws-c-compression：压缩算法（哈夫曼编码/解码）
- awslabs/aws-c-event-stream: 事件流消息处理（标头、前奏、有效载荷、crc/trailer）、事件流上的远程过程调用 (RPC) 实现
- awslabs/aws-c-http: C99 执行 HTTP /1.1 和 /2 规范 HTTP
- awslabs/aws-c-io: 套接字 (TCP、UDP) DNS、管道、事件循环、通道、SSL/TLS
- awslabs/aws-c-iot: C99 实现 AWS 物联网云服务与设备集成
- awslabs/aws-c-mqtt：适用于物联网 (IoT) 的标准轻量级消息传输协议
- awslabs/aws-c-s3: 用于与 Amazon S3 服务通信的 C99 库实现，旨在最大限度地提高高带宽 Amazon EC2 实例的吞吐量
- awslabs/aws-c-sdkutils: 用于解析和管理 AWS 配置文件的实用程序库
- awslabs/aws-checksums: 跨平台硬件加速，可CRC32c回退到CRC32高效的软件实现
- awslabs/aws-lc: 由 AWS 密码学团队根据来自 Google Boring 项目 AWS 和 Open SSL 项目的代码为其客户维护的通用密码库 SSL
- awslabs/s2n: C99 实施TLS/SSL协议，设计小巧快速，优先考虑安全性

除了 CRT Go 和 Rust SDKs 之外，其他所有版本都可用。

CRT依赖关系

这些CRT库构成了一个由关系和依赖关系组成的复杂网络。如果您需要CRT直接从源头构建，那么了解这些关系会很有帮助。但是，大多数用户通过自己的语言SDK（ AWS SDK例如 C++ 或 Java）或其语言物联网设备SDK（例如 AWS SDK AWS 适用于 C++ 的物联网或SDK适用SDK于 Java 的 AWS 物联网）来访问CRT功能。在下图中，“语言CRT绑定” 框指的是包装特定语言CRTSDK库的包。这

是形式为的软件包的集合`aws-crt-*`，其中 “*” 是一种SDK语言（例如`aws-crt-cpp`或 `aws-crt-java`）。

以下是CRT库的分层依赖关系的插图。

AWS SDK 和工具维护政策

概述

本文档概述了 AWS 软件开发套件 (SDK) 和工具（包括移动和物联网软件开发工具包）的维护政策及其底层依赖关系。 AWS 定期向 AWS SDK 和工具提供更新，其中可能包含对新增或更新 AWS 的 API、新功能、增强功能、错误修复、安全补丁或文档更新的支持。更新还可以解决依赖关系、语言运行时和操作系统的变化。 AWS SDK 版本发布给软件包管理器（例如 Maven NuGet、、PyPI），并作为源代码提供。 GitHub

我们建议用户继续 up-to-date 使用 SDK 版本，以了解最新功能、安全更新和底层依赖关系。不建议继续使用不受支持的 SDK 版本，但是否继续使用由用户自行决定。

版本控制

S AWS DK 发布版本采用 X.Y.Z 的形式，其中 X 代表主要版本。增加 SDK 的主版本表明该 SDK 进行了重大而实质性的更改，以支持该语言中的新习语和模式。当公共接口（例如类、方法、类型等）、行为或语义发生变化时，就会引入主要版本。应用程序需要更新才能使用最新的 SDK 版本。请务必根据AWS提供的升级指南谨慎更新主要版本。

SDK 主要版本生命周期

主要 SDK 和 Tools 版本的生命周期由 5 个阶段组成，概述如下。

- 开发者预览版（第 0 阶段）- 在此阶段，不支持 SDK，不应在生产环境中使用，并且仅用于抢先体验和反馈目的。未来版本可能会引入重大变更。一旦 AWS 确定某个版本为稳定产品，它就可以将其标记为候选版本。除非出现重大错误，否则候选版本已准备好发布，并且将获得全力 AWS 支持。
- 正式发布 (GA)（第 1 阶段）- 在此阶段，完全支持 SDK。 AWS 将提供常规的 SDK 版本，其中包括对新服务的支持、现有服务的 API 更新以及错误和安全修复。对于工具， AWS 将提供包含新功能更新和错误修复的常规版本。 AWS 将支持 GA 版本的 SDK 至少 24 个月。
- 维护公告（第 2 阶段）- AWS 将在 SDK 进入维护模式前至少 6 个月发布公告。在此期间，SDK 将继续得到全面支持。通常，维护模式是在下一个主要版本过渡到 GA 的同时宣布的。
- 维护（第 3 阶段）- 在维护模式期间， AWS 将 SDK 版本限制为仅解决关键错误修复和安全问题。SDK 不会收到新服务或现有服务的 API 更新，也不会更新以支持新区域。除非另有说明，否则维护模式的默认持续时间为 12 个月。

- 支持终止（第 4 阶段）- 当 SDK 达到支持终止时，它将不再接收更新或版本。之前发布的版本将继续通过公共包管理器提供，并且代码将保持不变 GitHub。 GitHub 存储库可能已存档。用户可以自行决定 end-of-support 是否使用已到达的 SDK。我们建议用户升级到新的主要版本。

以下是 SDK 主要版本生命周期的直观说明。请注意，下面显示的时间表仅供参考，不具约束力。

依赖生命周期

大多数 AWS SDK 都有底层依赖关系，例如语言运行时、操作系统或第三方库和框架。这些依赖项通常与语言社区或拥有该特定组件的供应商有关。每个社区或供应商都会发布自己的产品 end-of-support 时间表。

以下术语用于对底层第三方依赖项进行分类：

- 操作系统 (OS)：示例包括 Amazon Linux AMI、Amazon Linux 2、Windows 2008、Windows 2012、Windows 2016 等。
- 语言运行时系统：示例包括 Java 7、Java 8、Java 11、.NET Core、.NET Standard、.NET PCL 等。
- 第三方库/框架：示例包括 OpenSSL、.NET Framework 4.5、Java EE 等。

我们的政策是在社区或供应商终止对 SDK 依赖项的支持后至少 6 个月内继续支持 SDK 依赖项。但是，此策略可能会因具体的依赖项而有所不同。

> **Note**
>
> AWS 保留在不增加主要 SDK 版本的情况下停止对底层依赖项的支持的权利

沟通方式

维护公告将通过多种方式传达：

- 我们会向受影响的账户发送一封电子邮件公告，宣布我们计划终止对特定 SDK 版本的支持。该电子邮件将概述通往的路径 end-of-support，指定活动时间表，并提供升级指导。
- AWS SDK 文档（例如 API 参考文档、用户指南、SDK 产品营销页面和 GitHub 自述文件）已更新，以指明活动时间表并提供有关升级受影响应用程序的指导。

- 发布了一篇 AWS 博客文章，概述了通往该活动的路径 end-of-support，并重申了竞选时间表。
- 弃用警告已添加到 SDK 中，概述了 SDK 文档的路径 end-of-support 和链接。

要查看软件开发工具包和 AWS 工具的可用主要版本列表以及它们在维护生命周期中所处的位置，请参阅版本支持。

AWS SDKs和工具版本支持

下表显示了可用列表 AWS 软件开发套件 (SDK) 主要版本及其在维护生命周期中的位置，以及相关的时间表。有关主要版本生命周期的详细信息 AWS SDKs和 “工具” 及其底层依赖关系，请参阅维护政策。

SDK	主要版本	当前阶段	公开发行日期	注意
AWS CLI	1.x	公开发行	9/2/2013	
AWS CLI	2.x	公开发行	2020 年 10 月 2 日	
SDK对于 C++	1.x	公开发行	9/2/2015	
SDK适用于 Go V2	V2 1.x	公开发行	1/19/2021	
SDKfor Go	1.x	维护	11/19/2015	详情和日期请查看公告
SDK适用于 Java	1.x	维护	2010 年 3 月 25 日	详情和日期请查看公告
SDK适用于 Java	2.x	公开发行	11/20/2018	
SDK对于 JavaScript	1.x	停止支持	5/6/2013	
SDK对于 JavaScript	2.x	维护	6/19/2014	详情和日期请查看公告
SDK对于 JavaScript	3.x	公开发行	12/15/2020	
SDK对于 Kotlin 来说	1.x	公开发行	11/27/2023	
SDK对于。 NET	1.x	停止支持	11/2009	

SDK	主要版本	当前阶段	公开发行日期	注意
SDK对于。 NET	2.x	停止支持	11/8/2013	
SDK对于。 NET	3.x	公开发行	7/28/2015	
SDK for PHP	2.x	停止支持	11/2/2012	
SDK for PHP	3.x	公开发行	5/27/2015	
SDK适用于 Python (Boto2)	1.x	停止支持	7/13/2011	
SDK适用于 Python (Boto3)	1.x	公开发行	2015 年 6 月 22 日	
SDK适用于 Python (Botocore)	1.x	公开发行	2015 年 6 月 22 日	
SDK对于 Ruby	1.x	停止支持	7/14/2011	
SDK对于 Ruby	2.x	停止支持	2/15/2015	
SDK对于 Ruby	3.x	公开发行	8/29/2017	
SDK对于 Rust	1.x	公开发行	11/27/2023	
SDK为斯威夫特	1.x	公开发行	9/17/2024	
用于 PowerShell	2.x	停止支持	11/8/2013	
用于 PowerShell	3.x	停止支持	7/29/2015	
用于 PowerShell	4.x	公开发行	11/21/2019	

正在搜索未提及的SDK或工具？ 例如SDKs，加密SDKs、物联网设备和移动SDKs设备不包含在本指南中。要查找有关其他工具的文档，请参阅构建工具 AWS.

的文档历史记录 AWS SDKs和《工具参考指南》

下表描述了重要的新增内容和更新 AWS SDKs和《工具参考指南》。要获得有关本文档更新的通知，您可以订阅该订阅 RSS Feed。

变更	说明	日期
将 Swift SDK 添加到设置参考资料	为所有设置参考添加 Swift SDK 支持兼容性 AWS SDKs桌子。	2024年9月17日
SDK适用于 Java 1.x 系统属性	通过添加有关支持的JVM系统配置设置的详细信息 AWS SDK for Java 1.x。	2024 年 5 月 30 日
设置更新	添加JVM系统配置设置。	2024 年 3 月 27 日
兼容性表更新	更新了兼容性以获得SDK支持，更新了IAM身份中心程序。	2024 年 2 月 20 日
容器凭证更新。IMDS更新。	添加对 Amazon 的支持EKS。正在添加设置以禁用IMDSv1回退。	2023 年 12 月 29 日
请求压缩	正在为请求压缩特征添加设置。	2023 年 12 月 27 日
兼容性表	更新了兼容性表SDK和工具功能，包括 SDK Kotlin、SDK Rust 和 AWS Tools for PowerShell.	2023 年 12 月 10 日
身份验证更新	更新了SDKs和工具支持的身份验证方法。	2023 年 7 月 1 日

IAM最佳实践更新	更新了指南以符合IAM最佳实践。有关更多信息，请参阅中的安全最佳实践IAM。	2023 年 2 月 27 日
SSO更新	更新了新SSO令牌配置的SSO证书。	2022 年 11 月 19 日
设置更新	更新了常规配置和 Amazon S3 多区域接入点的支持表。	2022 年 11 月 17 日
设置更新	更新了IMDS客户端和IMDS凭据的清晰度。更新了环境变量。	2022 年 11 月 4 日
更新欢迎页面	宣布亚马逊 CodeWhisperer。	2022 年 9 月 22 日
更改单点登录的服务名称	更新以反映这一点 AWS SSO 现在被称为 AWS IAM Identity Center.	2022 年 7 月 26 日
设置更新	小幅更新配置文件详细信息和支持的设置。	2022 年 6 月 15 日
更新	大幅更新了本指南几乎所有部分。	2022 年 2 月 1 日
初始版本	本指南的第一版已向公众发布。	2020 年 3 月 13 日

AWS 术语表

有关最新的 AWS 术语，请参阅《AWS 词汇表参考》中的 AWS 词汇表。

本文属于机器翻译版本。若本译文内容与英语原文存在差异，则一律以英文原文为准。